This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ______________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 039 646 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
27.09.2000 Bulletin 2000/39

(51) Int. Cl.[7]: H03M 13/27

(21) Application number: 99400537.9

(22) Date of filing: 05.03.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant:
Mitsubishi Electric France
92000 Nanterre (FR)

(72) Inventor:
Belaiche, Vincent Antoine Victor
35500 VITRE (FR)

(74) Representative:
Blot, Philippe Robert Emile et al
c/o Cabinet Lavoix,
2, place d'Estienne d'Orves
75441 Paris Cedex 09 (FR)

(54) Interleaver device and method for interleaving a data set

(57) The interleaver device comprises a data processor (16) for running an interleaver (I^{-1}) under the control of driving means (18), input means (16) for inputting the data set to be interleaved and output means (18) for outputting the interleaved data set.

The driving means (18) include mapping processing means (20) for performing a set of bijective elementary functions (φ_n) and supplying a mapping of the interleaver to the data processor (16) for interleaving the data set according to this mapping and interleaver definition means (24) for supplying said mapping processing means, with a definition of said interleaver (I^{-1}), expressed as a compound function ($\varphi_k o \ldots o \varphi_1$) of elementary functions (φ_n), each elementary functions coming from said set of bijective elementary functions (φ_n), for said mapping processing means (20) to perform each of said functions, compounded according to the interleaver definition (I^{-1}), and so providing said mapping to the data processor (16).

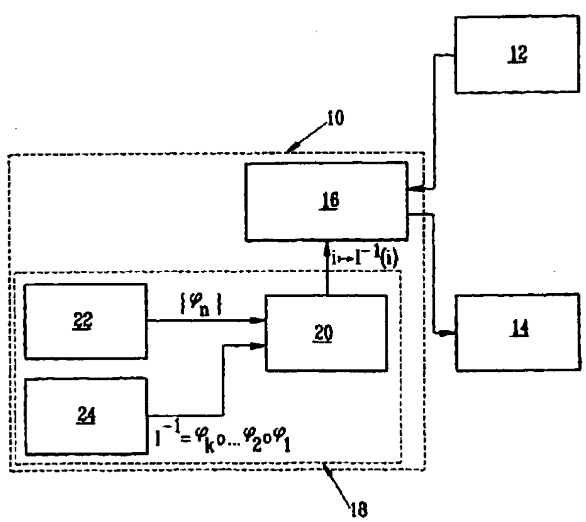


FIG.1

EP 1 039 646 A1

Printed by Xerox (UK) Business Services
2.16.7 (HRS)/3.6

09/10/2004, EAST Version: 1.4.1

Description

[0001] The invention concerns an interleaver device for interleaving a data set comprising a data processor for running an interleaver under the control of driving means and an input means for inputting the data set to be interleaved and output means for outputting the interleaved data set.
[0002] Such interleaver devices are specially useful in mobile telephones.
[0003] An interleaver is usually applied on informations transmitted on a radio channel between two radio stations.
[0004] In other technical fields, interleavers are also used for instance on data stored on a magnetic tape or a laser disk.
[0005] Turbo decoders have proved to offer near Shannon limit error protection in the case of white additive gaussian noise. One key feature of the Turbo encoder is an interleaver.
[0006] Interleaving data with a depth of N consists in:

- writing N consecutive symbols of data into a buffer of size of N symbols, i^{th} written symbol is written at position (i-1), and then
- read them in another order given by some permutation I of {0, ..., N-1}. i^{th} read symbol is read from position I(i-1).

[0007] In other words if i is the original position of a symbol in an input block, then $I^{-1}(i)$ is its position in the output block, where I^{-1} denotes the inverse permutation of I.
[0008] Deinterleaving data with a depth of N consists in:

- writing N consecutive symbols of data into a buffer of size of N symbols, i^{th} written symbol is written at position I(i-1), and
- then and then read them in another order given by some permutation I on {0, ..., N-1}. i^{th} read symbol is read from position i-1.

[0009] To implement the interleaver and the deinterleaver, function is needed that provides the mapping I. With some simple permutation I, this mapping can be computed by a simple analytical formula that can be quickly evaluated on any known per se processing means. For instance if the interleaver is a rectangular interleaver with L lines and C columns, we have:

$$\bullet\ N = L \cdot C$$

$$\bullet\ \forall i \in \{0,\ldots, N-1\}\ I^{-1}(i) = (i \text{ div } C) + (i \text{ mod } C) \cdot L \qquad (1)$$

[0010] In this formula (x div y) stands for the quotient of x divided by y in an Euclidian division and (x mod y) stands for the remainder in the same division.
[0011] The way this formula is to be understood is quite simple: a rectangular interleaver consists of an array of L lines and C columns. The input data is written along the lines and read along the column.
[0012] If l is line number (from 0 to L-1) and c is a column number (from 0 to C-1) of some written symbol then writing along the line comes to:

$$c = i \text{ mod } C \qquad (2)$$

$$l = i \text{ div } C \qquad (3)$$

that is to say:

$$i = l \cdot C + c \qquad (4)$$

l and c are the co-ordinate of the $(i+1)^{th}$ written symbol.
[0013] Reading along the columns comes to:

$$I^{-1}(i) = l + c \cdot L \qquad (5)$$

[0014] Note that as a matter of fact, equations (4) and (5) are similar when column and line roles are inverted. By substituting equations (3) and (2) in equation (5), we get equation (1) that shows that the I mapping can be obtained by simple arithmetic computations in the case of a rectangular interleaver.

09/10/2004, EAST Version: 1.4.1

[0015] However interleavers so simple as the rectangular interleaver do not always fit the needs of a forward error coding technique. In particular, in the case of the turbo encoding, the rectangular interleaver shows very poor performance. The interleaver to be used in the turbo encoder must be more random than the regular rectangular interleaver, and however not completely random as it must still keep some good properties of spacing interleaved symbols.

[0016] The simplest way to implement that kind of interleaver is to use a table where the i^{th} entry in the table is the value of $I(i-1)$. The size of the table is directly derived from N, that is to say that at least s(N) bits are needed to implement the table where:

$$s(x) = x \cdot \lceil \log_2(x) \rceil \qquad (6)$$

$\log_2(x)$ denotes the logarithm in base two of x, and $\lceil x \rceil$ denotes the ceiling of x, that is to say the least integer not less than x. When N grows bigger, then the amount of memory needed to store the table also grows bigger.

[0017] The aim of the invention is to provide an interleaver device which can run with very little memory to store the definition of the interleaver, even if the interleaver definition is very complex.

[0018] To this end, the invention relates to an interleaver device as defined in claim 1.

[0019] According to particular embodiments, the interleaver device may include one or more of the features defined in subclaims.

[0020] The invention proposes a method to build interleavers that are not so regular as the plain rectangular interleaver and nevertheless use far less memory than a completely tabulated interleaver.

[0021] In the case of a mobile telephone, due to real time constraint, it might be necessary to tabulate the interleaver of the invention to be able to use it, since processing the values might take too much time. However the definition can be very concise, thus holding little space in ROM. The interleaver table can held in RAM and computed off-line, during connection establishment. After call completion, the RAM can be reused for another purpose. Thus, even if the interleaver needs to be tabulated into RAM for effective use, this is all the same favourable, because the RAM can be used for something else when the call type does not need the interleaver.

[0022] Another advantage of having a concise definition from which a very large interleaver can be generated, is that a great variety of interleavers can be defined in the ROM memory, and only one of them selected at connection time.

[0023] In a particular embodiment, the interleaver definition could be transmitted over the air, as a parameter of the connection. The interleaver definition would have to follow the mobile telephone if during a hand over of the site where the interleaving is performed in the network is changed. Therefore, the next site would need to compute or select the table fast enough during the hand over preparation time.

[0024] The invention will be understood more clearly on reading the following description, given solely by way of example and made with reference to the drawings, in which :

- Fig. 1 is a block diagram of an interleaver device according to the invention;
- Fig. 2 is a flow-chart showing a Mod-threshold term splitting algorithm;
- Fig. 3 is an illustration of a term splitting function applied on a data set;
- Fig. 4 is an illustration of a Div-mod factor split function applied on a data set;
- Fig. 5 is an illustration of a mod-mod factor split function applied on a data set;
- Fig 6 is an illustration of a rectangular interleaver applied to a data set;
- Fig. 7 is an illustration of a truncated rectangular interleaver applied to a data set ; and
- Fig. 8 is an illustration of a puncturing of an interleaver.

[0025] The interleaver according to the invention can be implemented on any data processing means adapted to run under the driving of a software using the splitting of the interleaver as defined hereafter.

[0026] For example, the interleaver can be implemented in a mobile telephone.

[0027] Generally, and as shown on figure 1, the interleaver device includes a processing unit 10. It also includes means 12 for inputting a data set to be interleaved and means 14 for outputting the interleaved data.

[0028] The processing unit 10 includes a data processor 16 connected to the inputting means 12 for receiving a data set to be interleaved. It is also connected to outputting means 14 for supplying the interleaved data.

[0029] The data processor 16 is adapted for running an interleaved I^{-1}, the mapping of which is supplied by driving means 18. The driving means 18 comprise mapping processing means 20 which are adapted to compute a mapping of the interleaver from a set of elementary bijective functions φ_n which can be derived from a Read only memory 22.

09/10/2004, EAST Version: 1.4.1

The computation is performed, according to an interleaver definition I^{-1} received from interleaver definition means 24.

[0030] In the present embodiment, the elementary functions are stored in the Read only memory 22. Typically these elementary functions may be not completely defined in ROM 22, and zero one or more additional parameters may need to be supplied for them to be operated on their argument.

[0031] Interleaver definition means 24, typically consisting of a memory, are provided to supply the definition of the interleaver I^{-1} to mapping processing means 20. This definition is based on the elementary functions stored in memory 22. In particular, the interleaver I^{-1} is defined as a compound of the elementary functions φ_n stored in ROM 22, along with the suitable parameters if any. The mapping processing means 20 drive the data processor 10 for it to interleave data according to the mapping which is computed by the mapping processing means 20. This mapping is determined by performing each of the functions φ_n used to define the interleaver compounded according to the interleaver definition I^{-1}.

[0032] Typically the mapping processing means 20 compute the mapping from the definitions from ROM 22 and interleaver definition means 24 only once during connection establishment, and then keep it in a look-up table. In another embodiment, the mapping processing means 24 compute from the definition of I^{-1} the image $I^{-1}(i)$ of the position i of a data symbol to be interleaved on the fly at every symbol.

[0033] The implementation of the interleaver can be easily carried out by a man skilled in the art in view of the following explanations.

[0034] The definition of interleavers by the inventive method is recursive. The interleaver is defined by a permutation I^{-1} on the set {0, 1, ..., N-1} of the initial position indexes of the data to be interleaved. This set of position indexes, on which operates the interleaver I is split into a sum or a product of two smaller sets. Each of these two smaller sets can be then :

- either permuted by the underlying index permutation of a smaller interleaver,
- either split itself again in a sum or a product of two other smaller sets, or
- merged with a smaller set that was produced during a previous split of set.

[0035] The invention allows the use of interleavers external to the invention in place of the smaller interleavers. These external interleavers can be tabulated, or computed by a processing method different from that of the invention.

[0036] In the description, the following notations are used.

FINITE ELEMENTARY SET OF INTEGER

[0037] For any non null positive integer x, $\boxed{x}$ denotes the set:

$$\boxed{x} = \{i / \text{ i integer and } 0 \le i \le x-1\}$$

ELLIPSIS NOTATION

[0038] The ellipsis (...) notation is used in the middle of a sequence instead of the full sequence. This notation does not make any assumption on the number of literals in the formal sequence using it. These convention are quite obvious and usual. Below, some instances are given:

09/10/2004, EAST Version: 1.4.1

- $A_1 \times \ldots \times A_p$ Shall be interpreted just as A_1 if $p = 1$
- $A_1 \uplus \ldots \uplus A_p$ shall be interpreted just as A_1 if $p = 1$
- $A_1 \uplus \ldots \uplus A_p$ shall be interpreted just as the empty set $\varnothing$ if $p = 0$
- $\{x,x+1, \ldots, y-1,y\}$ where x and y are integers, shall be interpreted just as $\{x\}$ if $x=y$
- $\{x,x+1, \ldots, y-1,y\}$ where x and y are integers, shall be interpreted the empty set $\varnothing$ if $x>y$

PRODUCT OF SETS

[0039] If A and B are two sets, then their product A×B is the set of couples (a,b) such that a is in A and b in B. In other words:

$$A \times B = \{(a,b) / a \in A \text{ and } b \in B\}$$

[0040] The concept can be generalised to any finite number of sets:

$$A_1 \times \ldots \times A_p = \{(a_1,\ldots,a_p) / \forall i \; a_i \in A_i\}$$

[0041] Furthermore, for the sake of simplicity, A×(B×C), or (A×B)×C will just be denoted A×B×C, since there are obvious bijections between these three sets, that is to say there is little difference between ((a,b),c), (a,(b,c)) and (a,b,c).

[0042] One property of the product of sets is that if $A_1, A_2, \ldots A_p$ are finite sets and their respective numbers of elements are denoted $|A_1|, |A_2|, \ldots |A_p|$, then $A_1 \times \ldots \times A_p$ is also finite, and its number of elements $|A_1 \times \ldots \times A_p|$ is such that:

$$|A_1 \times \ldots \times A_p| = \prod_{i=1}^{i=p} |A_i|$$

UNION OF SETS

[0043] The union (noted $\cup$) of two sets is the set of elements that are in at least one of these two sets. That is to say $A \cup B = \{x / x \in A \text{ or } X \in B\}$.

DISJOINT UNION OF SETS (ALSO CALLED HERE SUM OF SETS)

[0044] If A and B are two sets, then their sum $A \uplus B$ is the set of couples (t,x) such that t is equal to 1 or 2 respectively when x is A or in B. Consequently,

$$A \uplus B = (\{1\} \times A) \cup (\{2\} \times B).$$

[0045] 1 and 2 are the usual integers and are used to distinguish the elements of A and of B in the union, so that the union is "disjoint".

[0046] The concept can be generalised to any finite number of sets:

09/10/2004, EAST Version: 1.4.1

$$A_1 \uplus \ldots \uplus A_p = (\{1\} \times A_1) \cup \ldots \cup (\{p\} \times A_p)$$

[0047] Where 1, 2, ...p are the usual whole numbers.

[0048] In the following for (t,x) in $A_1 \uplus \ldots \uplus A_p$, t is called the tag and x is called the value.

[0049] Furthermore, for the sake of simplicity, $A \uplus (B \uplus C)$, and $(A \uplus B) \uplus C$ will just be denoted $A \uplus B \uplus C$, since there are obvious bijections between these three sets, that is to say there is little difference between the tag sets {1, (2,1), (2,2)} , {1, (1,1), (1,2)},and {1,2,3}. The main use of the tag is its ability to distinguish the elements according to their set of origin.

[0050] Note that the disjoint union is generally defined in literature such that

$$A \uplus B = B \uplus A \text{ ,}$$

that is to say there is no underlying order on the set of tags, and therefore neither are the terms ordered. However the definition is slightly modified, because this property is not desirable for this application. In the following, the expression "sum of sets" instead of "disjoint union of sets" will be used.

[0051] Finally, one property of the disjoint union of sets is that if A_1, A_2, ... A_p are finite sets and their respective number of elements are denoted $|A_1|$, $|A_2|$, ...$|A_p|$, then $A_1 \uplus \ldots \uplus A_p$ is also finite, and its number of elements $|A_1 \uplus \ldots \uplus A_p|$ is such that:

$$|A_1 \uplus \ldots \uplus A_p| = \sum_{i=1}^{i=p} |A_i|$$

DISTRIBUTION OF PRODUCT OF SETS OVER SUM OF SETS.

[0052] In the following it is considered that:

$$A \times (B \uplus C) \text{ is the same as } (A \times B) \uplus (A \times C)$$

[0053] As a matter of fact, there is an obvious bijection mapping the element (a,(t,v)) of $A \times (B \uplus C)$ on the element (t,(a,v)) of

$$(A \times B) \uplus (A \times C).$$

This bijection just consists in placing the tag in first position.

[0054] Similarly it is considered that $(A \uplus B) \times C$ is the same as

[0055] The interleaver I is built as follows. It enables a concise definition of the interleaver using several bijective elementary functions.

[0056] The general idea to is to build the bijection I^{-1} $\mathbb{N} \rightarrow \mathbb{N}$ as the compound of several bijections $\varphi_1, \varphi_2, \ldots, \varphi_K$. That is to say $I^{-1} = \varphi_K \circ \ldots \circ \varphi_2 \circ \varphi_1$

[0057] This way we have:

09/10/2004, EAST Version: 1.4.1

$$\forall n \in \{1, \ldots k\}\ \varphi_n\ S^{(n-1)} \rightarrow S^{(n)}$$

With

$$S^{(0)} = \boxed{N}$$

and

$$S^{(k)} = \boxed{N}.$$

[0058] This way we have:

$$\begin{array}{lccccccc} I^{-1}: & \boxed{N} & & \longrightarrow & & & & \boxed{N} \\ & S^{(0)} & \xrightarrow{\varphi_1} & S^{(1)} & \xrightarrow{\varphi_2} & \cdots & \xrightarrow{\varphi_k} & S^{(k)} \\ & x & \longmapsto & \varphi_1(x) & \longmapsto & & \longmapsto & I^{-1}(x) = \varphi_k(\varphi_{k-1}(\ldots\ \varphi_2(\varphi_1(x))\ldots)) \end{array}$$

and I^{-1} is so built as a compound or sequence of k simpler steps, each step consisting of a bijective function φ_n. One more aspect is that each intermediate set $S^{(n)}$ can be defined as a sum of simpler sets $T_i^{(n)}$ that are products of sets of the $\boxtimes$ form:

$$\forall n \in \{1,\ldots,k\}\ S^{(n)} = T_1^{(n)} \uplus \ldots \uplus T_{q_n}^{(n)}$$

where :

$$\forall n \in \{1,\ldots,k\} \forall i \in \{1,\ldots,q_n\}\ T_i^{(n)} = \boxed{N_{i,1}^{(n)}} \times \boxed{N_{i,2}^{(n)}} \times \ldots \times \boxed{N_{i,p_{i,n}}^{(n)}}$$

and with (for all the φ_n are bijective), all the intermediate sets $S^{(n)}$ having N elements :

$$\forall n \in \{0,\ldots,k\},\ |S^{(n)}| = \sum_{i=1}^{i=q_n} \left(\prod_{j=1}^{j=p_{i,n}} N_{i,j}^{(n)} \right) = N$$

and also $q_0=1$ $p_{1,0} = 1$, $N_{1,1}^{(0)} = N$, $q_k = 1$, $p_{1,k} = 1$ and $N_{1,1}^{(k)} = N$, because

$$S^{(0)} = S^{(k)} = \boxed{N}.$$

[0059] Note that for the present application, it is needed that each φ_n be easily implementable by an algorithm that can be run on existing processing means.

[0060] Several types of φ_n elementary functions are defined. The different φ_n elementary functions form a set of elementary functions which are stored in memory 22, so that their algorithm can be run by the mapping processing means 20 according to the interleaver definition I^{-1}.

[0061] Here is an introductory list of the functions that will be presented in more details later:

A. Functions to simplify the definitions of the other functions by putting in first position the terms or factors on which the next function is to operate:

09/10/2004, EAST Version: 1.4.1

- Term permutation
- Factor permutation

B. Functions to decompose $S^{(n-1)}$ into an $S^{(n)}$ with more terms or factors:

- Elementary term split
- Elementary factor split

C. Functions ensuring the equivalence between $A\times(B\uplus C)$ and

$$(A\times B)\uplus(A\times C)$$

- Factorisation
- Distribution

D. Function to interleave a part of $S^{(n-1)}$

- Elementary interleaving: embedding of an interleaver external to the invention
- Affine-mod: a linear relation followed by a modulus, and operating on a term of $S^{(n-1)}$ as if it was a vector space whose dimension is the number of factors in this term,

E. Functions to simplify $S^{(n-1)}$ into an $S^{(n)}$ with fewer terms or factors:

- Elementary factor merge
- Elementary term merge

[0062] The different functions are as follows.

<u>TERM PERMUTATION φ_n</u>

[0063] In that case:

- $q_n = q_{n-1}$
- σ a permutation of $\{1,\ldots,q_n\}$, and
- $\forall i \in\{1,\ldots,q_n\} T^{(n)}_{\sigma(i)} = T^{(n-1)}_i$

φ_n is defined as:

$$\varphi_n: \begin{array}{ccc} S^{(n-1)} & \rightarrow & S^{(n)} \\ (t,v) & \mapsto & (\sigma(t),v) \end{array}$$

[0064] In the definition above t is the tag of (t,v), and v the value. Example:

$$S^{(n-1)} = \boxed{2}\times\boxed{3} \uplus \boxed{4}\times\boxed{5} \uplus \boxed{3}$$

$$S^{(n)} = \boxed{3} \uplus \boxed{4}\times\boxed{5} \uplus \boxed{2}\times\boxed{3}$$

$\sigma = (3,2,1)$

And for instance we have the following mappings:

$$\varphi_n: \quad \begin{array}{ccc} (1,(1,1)) & \mapsto & (3,(1,1)) \\ (3,(2)) & \mapsto & (1,(2)) \\ (2,(3,0)) & \mapsto & (2,(3,0)) \end{array}$$

<u>FACTOR PERMUTATION φ_n</u> The factor permutation φ_n function permutes the factors in the product constituting the first term of $S^{(n-1)}$. In fact the same principle could be used to permutes the factors of any term, but this is not needed for the definition thanks to the term permutation function defined above. In that case we have:

[0065]

- $q_n = q_{n-1}$
- $p_{1,n} = p_{1,n-1}$
- σ a permutation of $\{1,\ldots,p_{1,n-1}\}$
- $\forall i \in \{2,\ldots,q_n\}\ T_i^{(n)} = T_i^{(n-1)}$
- $\forall j \in \{1,\ldots,p_{1,n-1}\}\ N_{1,j}^{(n)} = N_{1,\sigma(j)}^{(n-1)}$

φ_n is defined as:

$$\varphi_n: \quad \begin{array}{ccc} S^{(n-1)} & \rightarrow & S^{(n)} \\ x = (t, (x_1, x_2 \ldots x_{p_{t,n-1}})) & \mapsto & \begin{cases} (1, (x_{\sigma(1)}, x_{\sigma(2)} \ldots x_{\sigma(p_{1,n-1})})) & \text{if } t=1 \\ x & \text{otherwise} \end{cases} \end{array}$$

Example:

$$S^{(n-1)} = \boxed{2}\times\boxed{3}\times\boxed{4} \uplus \boxed{3}$$

$$S^{(n-1)} = \boxed{3}\times\boxed{2}\times\boxed{4} \uplus \boxed{3}$$

$\sigma = (2,1,3)$

And for instance we have the following mappings:

$$\varphi_n: \quad (1,(1,2,3)) \mapsto (1,(2,1,3))$$
$$(2,(2)) \mapsto (2,(2))$$
$$(1,(0,1,2)) \mapsto (1,(1,0,2))$$

[0066] For the term or factor permutation, φ_n is made at null processing cost. The only added value of φ_n is that it makes simpler the definitions that will follow by alleviating the notation. The following definitions are applicable to any terms or factors in a sum or product of sets, respectively, instead, thanks to the factor permutation φ_n we can without any loss of generality make them only for the first or the first and some subsequent terms or factors according to the case.

[0067] In consequence the permutation σ is typically the compound of zero, one or two transpositions. A transposition is a permutation that swaps two elements of a set, and let the other unchanged. Thus such a permutation σ puts in the first position the terms or factors on which the next function φ_n is to operate. In other words the term or factor permutation function specifies on which term(s) or factor(s) is operating the next function in the $\varphi_k \circ \ldots \circ \varphi_2 \circ \varphi_1$ compound.

ELEMENTARY TERM SPLIT φ_n

[0068] The elementary term split function φ_n is splitting the first factor in the first term of $S^{(n-1)}$ into a $\uplus$ sum of two sets. In fact, the same principle could be used to split any factor of any term, but this is not needed for the definition, thanks to the term and factor permutation functions defined above. An example is given on Figure 3.

[0069] $S^{(n)}$ is such that from $S^{(n-1)}$ to $S^{(n)}$, nothing is changed except the first factor $\boxed{A+B}$ of the first term $T_1^{(n-1)}$ that is split by a bijection f into the $\uplus$ sum of $\boxed{A}$ and $\boxed{B}$ on which are then distributed the subsequent factors, if any, of the first term of $S^{(n-1)}$.

[0070] In that case :

- $q_n = q_{n-1}+1$, i.e. one more term
- $\forall i \in \{2,\ldots,q_{n-1}\}\ T_{i+1}^{(n)} = T_i^{(n-1)}$, i.e. the subsequent terms unchanged
- A and B are two positive and non null integers, and $N_{1,1}^{(n-1)} = A+B$, $N_{1,1}^{(n)} = A$ and $N_{2,1}^{(n)} = B$
- $p_{1,n} = p_{2,n} = p_{1,n-1}$ and $\forall j \in \{2,\ldots, p_{1,n-1}\}\ N_{1,j}^{(n)} = N_{2,j}^{(n)} = N_{1,j}^{(n-1)}$
- a bijection

$$f: \quad \boxed{A+B} \rightarrow \boxed{A} \uplus \boxed{B}$$
$$x \mapsto (t_f(x), v_f(x))$$

where $t_f(x)$ and $v_f(x)$ respectively denote the tag and the value of f(x) in $\boxed{A} \uplus \boxed{B}$.

[0071] Note that the conditions of 3rd and 4th bullets are such that :

$$T_1^{(n-1)} = \boxed{A+B} \times T'$$

09/10/2004, EAST Version: 1.4.1

$$T_1^{(n)} = \boxed{A} \times T'$$

$$T_2^{(n)} = \boxed{B} \times T'$$

$$\text{And } T' = \boxed{N_{1,2}^{(n-1)}} \times \ldots \times \boxed{N_{1,p_{1,n-1}}^{(n-1)}}$$

φ_n is built as:

$$\varphi_n: S^{(n-1)} \to S^{(n)}$$

$$x \mapsto \begin{cases} (t_f(x_1), (v_f(x_1), x_2 \ldots x_{p_{1,n-1}})) & \text{if } t=1 \\ (t+1, v) & \text{otherwise} \end{cases}$$

with

$$x = (t,v)$$

and

$$v = (x_1, x_2 \ldots x_{p_{t,n-1}})$$

[0072] The f bijections used by the elementary term split function can be defined as follows.

$$\underline{f : \boxed{A+B} \to \boxed{A} \uplus \boxed{B} \text{ main functions}}$$

[0073] There are many possible functions that all are computable on any current processor.

[0074] As an example of f, the mod-threshold term split function will be described hereafter:

[0075] In this function, the modulus of an element x of $\boxed{A+B}$ by some constant C is computed. Then it is compared to a threshold T. According to the result of this comparison, it is decided to map x either to $\boxed{A}$ or to $\boxed{B}$. Constants C and T are such that:

$$0 < T < C$$

[0076] In order to understand more easily how the mod-threshold function works, a simplified algorithm will be first presented. The simplified algorithm cannot make the mapping in a random way, but only in a sequential way, that is to say it can map element x, only after having mapped elements 0, 1, ...,x-1 before.

[0077] In the simplified algorithm there are two counters n_A and n_B that hold the number of elements that have already been respectively mapped to $\boxed{A}$ and $\boxed{B}$.

09/10/2004, EAST Version: 1.4.1

```
nA := 0;
nB := 0;
for x:= 0 to A+B-1 do
        begin
            if (nA = A) then
              begin
                 remark   [A] is full, then map x to [B]
                 map x to (2,nB);
                 nB := nB + 1;
              end
            else if (nB = B) then
              begin
                 remark   [B] is full, then map x to [A]
                 map x to (1,nA);
                 nA := nA + 1;
              end
          else
              begin
                 remark   neither [A] or [B] are yet full, then map x accord-
              ing to threshold of modulus
                 c := x mod C;
                 if c<T then
                      begin
```


09/10/2004, EAST Version: 1.4.1

```
                    map x to (1,n_A);
                    n_A := n_A + 1;
                end
            else
                begin
                    map x to (2,n_B);
                    n_B := n_B + 1;
                end
        end
    end
```

[0078] Now, an algorithm that can work with random values of x in input (x not given in a sequential way) is presented.

[0079] Three constants x_M, v_M and t_M are derived from A, B, C and T according to the following definitions and formulas: x_M is the greatest x while neither $\boxed{A}$ nor $\boxed{B}$ are yet full in the simplified algorithm above.

t_M is the tag of the set ($\boxed{A}$ or $\boxed{B}$) not yet full after x_M has been reached by x in the simplified algorithm
v_M is the greatest value of the set not yet full while both sets are not yet full in the simplified algorithm.
x_M, t_M and v_M can be alternatively defined by the following formulas:

$$x_M \triangleq \max\left\{ x \in \boxed{A+B} \middle/ \begin{pmatrix} l \cdot T + c < A & \text{if } c < T \\ \text{and} & \\ l \cdot (C-T) + c - T < B & \text{otherwise} \end{pmatrix}, \text{ where } l \triangleq x \text{ div } C, \text{ and } c \triangleq x \bmod C \right\}$$

$$t_M \triangleq \begin{pmatrix} 1 & \text{if } ((x_M+1) \bmod C) \geq T \\ 2 & \text{otherwise} \end{pmatrix}$$

$$\text{if } t_M = 1 \text{ then } v_M \triangleq \max\left\{ v \middle/ \exists x \leq x_M, \begin{pmatrix} l \triangleq x \text{ div } C \\ \text{and} \\ c \triangleq x \bmod C \end{pmatrix} \Rightarrow \begin{pmatrix} c < T \\ \text{and} \\ v = l \cdot T + c \end{pmatrix} \right\}$$

$$\text{if } t_M = 2 \text{ then } v_M \triangleq \max\left\{ v \middle/ \exists x \leq x_M, \begin{pmatrix} l \triangleq x \text{ div } C \\ \text{and} \\ c \triangleq x \bmod C \end{pmatrix} \Rightarrow \begin{pmatrix} c \geq T \\ \text{and} \\ v = l \cdot (C-T) + c - T \end{pmatrix} \right\}$$

09/10/2004, EAST Version: 1.4.1

[0080] Then the algorithm that maps x onto (t,v) is the following:

$$f: \quad \boxed{A+B} \rightarrow \boxed{A} \uplus \boxed{B}$$

$$x \longmapsto (t,v)$$

defined by the algorithm below

```
if (x ≤ x_M) then
    begin
        remark Neither [A] nor [B] are yet full, then discrimination is based
                on threshold
        if (x mod C) < T then
            begin
                t := 1;
                v := (x div C)·T + (x mod C);
            end
        else
            begin
                t := 2;
                v := (x div C)·(C-T) + ((x mod C)-T);
            end
    end
else
    begin
        remark [A] or [B] is already full, then no more discrimination, the
                remainder of [A+B] is mapped to the set not yet full, that is
                to say that with tag t_M
        t := t_M;
        v := x - x_M + v_M;
    end
```

[0081] The Mod-threshold term splitting algorithm is also shown on Figure 2. The result of the algorithm is illustrated on figure 3 in the case of A=5, B=10, C=4 and T=2.

[0082] For some particular value to T and C the definition can be simplified. For instance when C = A+B and T = A, f is defined as:

09/10/2004, EAST Version: 1.4.1

$$
\begin{aligned}
f:\quad & \boxed{A+B} && \rightarrow && \boxed{A}\uplus\boxed{B} \\
& x && \mapsto && (t,v) \\
&&&&& \text{with if } x < A \\
&&&&& \qquad \text{then } t = 1 \text{ and } v = x \\
&&&&& \qquad \text{else } t = 2 \text{ and } v = x - A
\end{aligned}
$$

[0083] Another simplification occurs when given the values of D=A+B, C and T , A and B are chosen such that the comparison of x to x_M is not necessary, and only the comparison of the modulus to the threshold needs to be done. That is to say the mod-threshold term split function can be more simply written as:

$$
\begin{aligned}
f:\quad & \boxed{A+B} && \rightarrow && \boxed{A}\uplus\boxed{B} \\
& x && \mapsto && (t,v) \\
&&&&& \text{with if } c < T \\
&&&&& \text{then } t = 1 \text{ and } v = l\cdot T + c \\
&&&&& \text{else } t = 2 \text{ and } v = l\cdot(C-T) + c - T \\
&&&&& \text{where} \begin{cases} c = x \bmod C \\ l = x \text{ div } C \end{cases}
\end{aligned}
$$

[0084] Given D = A+B, C and T , the value of A and B for which this simplification occurs can be computed as:

$$
\begin{cases}
A = \left|\left\{x \in \boxed{D} \,/\, (x \bmod C) < T\right\}\right| \\
B = \left|\left\{x \in \boxed{D} \,/\, (x \bmod C) \geq T\right\}\right|
\end{cases}
\tag{7}
$$

where |.| denotes the number of elements in a set.

[0085] This can also be more simply computed as:

$$
\begin{cases}
A = 1 + l_A\cdot T + c_A \quad \text{where} \begin{cases} l_A = x_A \text{ div } C, \\ c_A = x_A \bmod C, \text{ and} \\ x_A = \max\left\{x \in \boxed{D} \,/\, (x \bmod C) < T\right\} \end{cases} \\
B = 1 + l_B\cdot(C-T) + c_B - T \quad \text{where} \begin{cases} l_B = x_B \text{ div } C, \\ c_B = x_B \bmod C, \text{ and} \\ x_B = \max\left\{x \in \boxed{D} \,/\, (x \bmod C) \geq T\right\} \end{cases}
\end{cases}
$$

09/10/2004, EAST Version: 1.4.1

[0086] Finally x_A and x_B can be more simply computed by the following algorithm:

```
c := (D-1) mod C;
if c < T then
    begin
        xA := (D-1);
        xB := (D-1) – (c+1);
    end
else
    begin
        xB := (D-1);
        xA := (D-1) – (c+1) + T;
    end
```

[0087] For instance D=10 C=5 T=2 will yield $x_B = 9$, $x_A = 6$, $l_A = 1$, $c_A = 1$, $l_B = 1$, $c_B = 4$, A = 4 and B = 6.
[0088] Selecting A and B such that the algorithm is simplified can result in less processing power requirement to prepare to interleaver table, or to compute the interleaver real time, or less Integrated Circuit surface, if this real time term splitting procedure is performed by an ASIC.

ELEMENTARY FACTOR SPLIT φ_n

[0089] The elementary factor split φ_n function is splitting the first factor in the first term of $S^{(n-1)}$ into a × product of two sets. In fact the same principle could be used to split any factor of any term, but this is not needed for the definition thanks to the term and factor permutation functions defined above.
[0090] $S^{(n)}$ is such that from $S^{(n-1)}$ to $S^{(n)}$, nothing is changed except the first factor $\boxed{A \cdot B}$ of the first term that is split by a bijection g into the × product of $\boxed{A}$ and $\boxed{B}$.
[0091] In that case:

- $q_n = q_{n-1}$, i.e. same number of terms
- $\forall i \in \{2,\ldots,q_{n-1}\}\ T_i^{(n)} = T_i^{(n-1)}$, i.e. the subsequent terms unchanged
- $p_{1,n} = p_{1,n-1}+1$, i.e. one more factor in first term
- $\forall j \in \{2,\ldots,p_{n-1}\}\ N_{1,j+1}^{(n)} = N_{1,j}^{(n-1)}$
- A and B defined by $A \triangleq N_1^{(n)}$, $B \triangleq N_2^{(n)}$, and thus $A \cdot B = N_1^{(n-1)}$
- a bijection

$$g: \quad \boxed{A \cdot B} \to \boxed{A} \times \boxed{B}$$
$$\quad x \mapsto (g_1(x), g_2(x))$$

φ_n is built as:

09/10/2004, EAST Version: 1.4.1

$$\varphi_n: \quad S^{(n-1)} \rightarrow S^{(n)}$$

$$x \mapsto \begin{cases} (1,(g_1(x_1), g_2(x_1), x_2 \ldots, x_{p_{1,n-1}})) & \text{if } t=1 \\ x & \text{otherwise} \end{cases}$$

with

$$x = (t,v)$$

and

$$v = (x_1, x_2 \ldots x_{p_{1,n-1}})$$

[0092] The g bijections used by the elementary factor split function can be defined as follows:

<u>$g: \boxed{A \cdot B} \rightarrow \boxed{A} \times \boxed{B}$ main factor split functions</u>

[0093] There are many possible functions that all are computable on any current processor.
[0094] Here are some examples:
[0095] In the following for all x and y integer such that y>0:

- x div y stands for the quotient of Euclidian division of x by y,
- x mod y stands for the remainder of Euclidian division of x by y.

[0096] Note that $\forall x\ \forall y\ y>0$, $(x \bmod y) \in \boxed{y}$, even for $x \leq 0$
[0097] The DIV-MOD factor split function:

$$g: \quad \boxed{A \cdot B} \rightarrow \boxed{A} \times \boxed{B}$$

$$x \mapsto (x \text{ div } B, x \text{ mod } B)$$

[0098] This function is clearly bijective and:

$$g^{-1}((a,b)) = a \cdot B + b$$

[0099] The div-mod factor split function is illustrated on Figure 4.
[0100] The MOD-MOD factor split function:
This function can be used only when the greatest common divider of A and B is 1 (that is to say $A \wedge B = 1$).

$$g: \quad \boxed{A \cdot B} \rightarrow \boxed{A} \times \boxed{B}$$

$$x \mapsto (x \text{ mod } A, x \text{ mod } B)$$

[0101] The fact that the greatest common divider of A and B is 1 ensures the existence of two integers A' and B'

09/10/2004, EAST Version: 1.4.1

such that $B \cdot B' + A \cdot A' = 1$.

[0102] This allows to build easily the inverse of g as: $g^{-1}((a,b)) = b + (((a-b) \cdot B') \bmod A) \cdot B = (a \cdot B \cdot B' + b \cdot A \cdot A') \bmod (A \cdot B)$

[0103] The mod-mod factor split function is illustrated on Figure 5.

FACTORISATION φ_n

[0104] The factorisation φ_n function groups several equal terms of a sum of sets, into a product of sets. This uses the obvious bijection between:

$$\underbrace{B \uplus \ldots \uplus B}_{\text{A times}} \qquad \text{And} \qquad \boxed{A} \times B$$

[0105] This bijection simply maps the tag-value couple (t,v) on the value only list (t-1,v), making of the tag t an element t-1 of $\boxed{A}$.

[0106] The factorisation φ_n thus factorises the A first terms for $S^{(n-1)}$. In fact the same principle could be used to factorise any terms in $S^{(n-1)}$ provided that they are equal, but this is not needed for the definition thanks to the term permutation functions defined above.

[0107] $S^{(n)}$ and $S^{(n-1)}$ are such that:

- A is positive non null integer such that $q_n = q_{n-1} - A + 1$, because A terms of $S^{(n-1)}$ are merged in one,
- $\forall i \in \{1, \ldots, A\}\ T_i^{(n-1)} = T_1^{(n-1)}$, that is to say the A first terms of $S^{(n-1)}$ are identical,
- $\forall i \in \{2, \ldots, q_n\}\ T_i^{(n)} = T_{i+A-1}^{(n-1)}$, that is to say the terms following the A first terms of $S^{(n-1)}$ are unaffected

φ_n is built as:

$$\varphi_n: \quad S^{(n-1)} \rightarrow S^{(n)}$$

$$x \mapsto \begin{cases} (x_1 + 1, (x_2, x_3 \ldots, x_{p_{1,n-1}})) & \text{if } t=1 \\ (t+A-1, v) & \text{otherwise} \end{cases}$$

with

$x = (t,v)$

and

$v = (x_1, x_2 \ldots x_{p_{t,n-1}})$

DISTRIBUTION φ_n

[0108] The distribution φ_n function does the inverse operation of the factorisation $\varphi'n$ that was described in the previous section. That is to say the first factor of the first term in $S^{(n-1)}$ becomes a tag for a sum.

[0109] In fact the same principle could be used on any factor of any term of $S^{(n-1)}$ provided that the term containing that factor has at least two factors, but this is not needed for the definition thanks to the term permutation functions defined above.

[0110] $S^{(n)}$ and $S^{(n-1)}$ are such that:

09/10/2004, EAST Version: 1.4.1

- A is a positive non null integer such that $N_{1,1}^{(n-1)} = A$
- $q_n = q_{n-1} + A-1$, that is to say 1 term of $S^{(n-1)}$ is distributed over A terms of $S^{(n)}$
- $p_{1,n-1} > 1$ (first term of $S^{(n-1)}$ has at least two factors)
- $\forall i \in \{1, \ldots, A\}\ p_{i,n} = p_{1,n-1} - 1$, the A first terms of $S^{(n)}$ have one factor fewer - the one that was distributed - than the first term of $S^{(n-1)}$
- $\forall i \in \{1,\ldots,A\} \forall j \in \{1,\ldots,p_{1,n}-1\}\ N_{i,j}^{(n)} = N_{1,j+1}^{(n-1)}$, the A first terms of $S^{(n)}$ are all identical and equal to the first term of $S^{(n-1)}$ without its first factor
- $\forall i \in \{2,\ldots, q_{n-1}\}\ T_{i+A-1}^{(n)} = T_i^{(n-1)}$, the q_{n-1}-1 subsequent terms of $S^{(n-1)}$ are unaffected

φ_n is built as:

$$\varphi_n: \quad S^{(n-1)} \to S^{(n)}$$

$$x \mapsto \begin{cases} (x_1+1, (x_2, x_3 \ldots, x_{p_{1,n-1}})) & \text{if } t=1 \\ (t+A-1, v) & \text{otherwise} \end{cases}$$

with

$x = (t,v)$

and

$v = (x_1, x_2 \ldots x_{p_{1,n-1}})$

<u>ELEMENTARY INTERLEAVING φ_n</u>

[0111] The elementary interleaving φ_n function permutes the elements of the first factor in the first term of $S^{(n-1)}$. In fact the same principle could be used to permute the elements of interleave any factor of any term, but this is not needed for the definition thanks to the term and factor permutation functions defined above.

[0112] $S^{(n)}$ and $S^{(n-1)}$ are of course equal.

[0113] In that case:

- $S^{(n)} = S^{(n-1)}$
- A an integer such that $A = N_{1,1}^{(n)}$
- a bijection $\iota : \boxed{A} \to \boxed{A}$ φ_n is built as:

$$\varphi_n: \quad S^{(n-1)} \to S^{(n)}$$

$$x \mapsto \begin{cases} \left(1, (\iota(x_1), x_2 \ldots, x_{p_{1,n-1}})\right) & \text{if } t=1 \\ x & \text{otherwise} \end{cases}$$

with

$x = (t,v)$

and

$v = (x_1, x_2 \ldots x_{p_{1,n-1}})$

[0114] Here are some examples of ι bijection used by the elementary permutation function.

[0115] The tabulated ι function:

09/10/2004, EAST Version: 1.4.1

[0116] In that case the ι is implemented by a table in the memory of the processing unit. There are A entries in that table and ι (x) is written in the $(x+1)^{th}$ entry.

[0117] The table therefore requires s(A) bits (where s is defined in equation (6))

AFFINE-MOD φ_n

[0118] The affine-mod φ_n function acts in the first terms of $S^{(n-1)}$ as if it was a vector space of dimension $p_{1,n-1}$. In fact the same principle could be used on any term, but this is not needed for the definition thanks to the term permutation functions defined above.

[0119] $S^{(n)}$ and $S^{(n-1)}$ are of course equal.

[0120] In that case:

- $S^{(n)} = S^{(n-1)}$
- $m \triangleq p_{1,n-1}$
- an m line m column matrix $U=[u_{i,j}]$ such that

$$\forall i \in \{1,\ldots,m\} \forall j \in \{1,\ldots,m\}\ u_{i,j} \text{ is integer (possibly negative)}$$

- a m line vector $V=[v_i]$ such that:

$$\forall i \in \{1,\ldots,m\} v_i \text{ is integer (possibly negative)}$$

φ_n is built as:

$$\varphi_n: \quad S^{(n-1)} \rightarrow S^{(n)}$$

$$(t,x) \mapsto \begin{cases}(1,y) \text{ if } t = 1\\(t,x) \text{ otherwise}\end{cases}$$

with for t=1

$$x = (x_1, x_2, \ldots, x_m)$$

$$y = (y_1, y_2, \ldots, y_m)$$

such that:

$$\begin{bmatrix} z_1 \\ \cdot \\ \cdot \\ z_m \end{bmatrix} = U \cdot \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_m \end{bmatrix} + V$$

and

$$\forall i \in \{1,\ldots,m\} y_i = z_i \bmod N_i^{(n)}$$

[0121] Furthermore U and V are such that the φ_n obtained be bijective.

[0122] One sufficient condition to be fulfilled by U and V for the affine-mod φ_n to be bijective is the following one: U is the product of two matrices $U^{(1)}$, and $U^{(2)}$.

- The elements of these matrices are denoted this way: $\forall k \in \{1,2\}\ U^{(k)} = [u_{i,j}^{(k)}]$

- $U^{(1)}$ is a sub-diagonal matrix with a diagonal of 1 or -1, that is to say:

$$\begin{cases}\forall\ (i,j)\in\{1,\dots,m\}^2,\ j>i \Rightarrow u^{(1)}_{i,j}=0\\ \text{and}\\ \forall\ i\in\{1,\dots,m\},\quad u^{(1)}_{i,i}=1 \text{ or } u^{(1)}_{i,i}=-1\end{cases}$$

- $U^{(2)}$ is a diagonal matrix with for all i, i^{th} diagonal element primary with $N^{(n)}_{1,i}$,

[0123] That is to say:

$$\begin{cases}\forall\ (i,j)\in\{1,\dots,m\}^2,\ j\neq i \Rightarrow u^{(2)}_{i,j}=0\\ \text{and}\\ \forall\ i\in\{1,\dots,m\},\quad u^{(2)}_{i,i} \text{ and } N^{(n)}_{1,i} \text{ greatest common divider is 1}\end{cases}$$

- Product is done in this order: $U = U^{(2)} \cdot U^{(1)}$

Note: $U^{(1)}$ can be also super-diagonal (elements under diagonal are null) instead of sub-diagonal, this comes up to the same thanks to the factor permutation φ_n function.

ANNEXE ON AFFINE-MOD FUNCTIONS

[0124] Here is given a proof of the sufficient condition that was given to build the affine-mod φ_n functions. Let us recall the definition of the affine-mod function where we have omitted the n index that brings no information here, and we have replaced it by a U index, meaning that this is the φ that is generated by some matrix U. For alleviating the notation we have also omitted the term index (specifying that we are working on the first term of $S^{(n-1)}$).

$$\varphi_U:\quad \boxed{N_1}\times\boxed{N_2}\times\dots\times\boxed{N_p} \;\rightarrow\; \boxed{N_1}\times\boxed{N_2}\times\dots\times\boxed{N_p}$$

$$(x_1, x_2 \dots x_p) \;\mapsto\; (y_1, y_2 \dots, y_p)$$

such that:

$$\begin{bmatrix} z_1\\ \cdot\\ \cdot\\ z_m\end{bmatrix} = U\cdot\begin{bmatrix} x_1\\ \cdot\\ \cdot\\ x_p\end{bmatrix} + V$$

and

$$\forall i\in\{1,\dots,p\}\, y_i = z_i \bmod N_i$$

[0125] Since the from and to sets are the same and are finite, it is clearly necessary and sufficient to show that φ_U is injective in order to show that φ_U is bijective.

[0126] Let **Z** denote the set of signed integers, then bijectivity of φ_U is equivalent to the injectivity definition, i.e.:

09/10/2004, EAST Version: 1.4.1

$$\left.\begin{array}{l}\forall(q_1, q_2 \ldots q_p) \in \mathbf{Z}^p \\ \forall(x_1, x_2 \ldots x_p) \in \boxed{N_1} \times \boxed{N_2} \times \ldots \times \boxed{N_p} \\ \forall(y_1, y_2 \ldots y_p) \in \boxed{N_1} \times \boxed{N_2} \times \ldots \times \boxed{N_p}\end{array}\right\}, U \cdot \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_p \end{bmatrix} + V = U \cdot \begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ y_p \end{bmatrix} + V + \begin{bmatrix} q_1 \cdot N_1 \\ \cdot \\ \cdot \\ q_p \cdot N_p \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_p \end{bmatrix} = \begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ y_p \end{bmatrix}$$

[0127] Then by a simple subtraction, by using linearity of U, and by making some variable change on x_i-y_i it comes that bijectivity of φ_U is equivalent to

$$\left.\begin{array}{l}\forall(q_1, q_2 \ldots q_p) \in \mathbf{Z}^p \\ \forall(x_1, x_2 \ldots x_p) \in [-N_1+1\ ,\ N_1-1] \times \ldots \times [-N_p+1\ ,\ N_p-1]\end{array}\right\}, U \cdot \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_p \end{bmatrix} = \begin{bmatrix} q_1 \cdot N_1 \\ \cdot \\ \cdot \\ q_p \cdot N_p \end{bmatrix} \Rightarrow \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_p \end{bmatrix} = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

[0128] Where in the equation above all [a,b] denotes an interval of signed integers comprised between a and b inclusive.

[0129] Now let us assume that U is a sub-diagonal matrix, with a diagonal of 1 or -1, that is to say:

$$\begin{cases} \forall\ (i,j) \in \{1,\ldots,m\}^2, j>i \Rightarrow u_{i,j} = 0 \\ \text{and} \\ \forall\ i \in \{1,\ldots,m\}, \quad u_{i,i} = 1 \text{ or } u_{ii} = -1 \end{cases}$$

(with of course all the elements of U integers).

[0130] If such a condition is fullfilled then U is clearly inversible and the inverse U^{-1} of U is also a sub-diagonal matrix with a diagonal of 1 or -1 and all elements still integers. This is trivially derived by the Gauss pivot algorithm applied to inverse matrix U.

[0131] The equation

$$U \cdot \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_p \end{bmatrix} = \begin{bmatrix} q_1 \cdot N_1 \\ \cdot \\ \cdot \\ q_p \cdot N_p \end{bmatrix}$$

can be written:

$$\begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_p \end{bmatrix} = U^{-1} \cdot \begin{bmatrix} q_1 \cdot N_1 \\ \cdot \\ \cdot \\ q_p \cdot N_p \end{bmatrix}$$

[0132] Let us denote $u'_{i,j}$ the elements of $U^{-1}=[u'_{i,j}]$

[0133] U^{-1} is a subdiagonal matrix with all element integer and a diagonal of 1 or -1.

09/10/2004, EAST Version: 1.4.1

[0134] Then by recurrence:

- x_1 is clearly null, because $x_1 = u'_{1,1} \cdot q_1 \cdot N_1$, q_1 and $u'_{1,1}$ are integer, and 0 is the only multiple N_1 in $[-N_1+1, N_1-1]$
- Now for all j with $1 \le j < p$, if vi $1 \le i \le j$, $x_i = 0$, then we have $x_{j+1} = u'_{j+1,j+1} \cdot q_{j+1} \cdot N_{j+1}$, and similarly we can deduce that x_{j+1} is also null.

[0135] Now let us assume that $U=U^{(2)} \cdot U^{(1)}$, where $U^{(1)}$ is a subdiagonal matrix with all elements integers and a diagonal of 1 or -1, and $U^{(2)}$ is a diagonal matrix whose for all i, i^{th} diagonal element is primary with N_i.

[0136] Let us assume that for some $(q_1, \ldots, q_p)$ integers, and for some $(x_1, \ldots x_p)$ in $[-N_1+1, N_1-1] \times \ldots \times [-N_p+1, N_p-1]$ we have:

$$U \cdot \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_p \end{bmatrix} = \begin{bmatrix} q_1 \cdot N_1 \\ \cdot \\ \cdot \\ q_p \cdot N_p \end{bmatrix}$$

[0137] Let us call

$$\begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ y_p \end{bmatrix} = U^{(1)} \cdot \begin{bmatrix} x_1 \\ \cdot \\ \vdots \\ x_p \end{bmatrix}$$

we have therefore:

$$U^{(2)} \cdot \begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ y_p \end{bmatrix} = \begin{bmatrix} q_1 \cdot N_1 \\ \cdot \\ \cdot \\ q_p \cdot N_p \end{bmatrix}$$

[0138] That is to say:

$$\begin{bmatrix} u^{(2)}_{1,1} \cdot y_1 \\ \cdot \\ \cdot \\ u^{(2)}_{p,p} \cdot y_p \end{bmatrix} = \begin{bmatrix} q_1 \cdot N_1 \\ \cdot \\ \cdot \\ q_p \cdot N_p \end{bmatrix}$$

[0139] For all i from 1 to m, N_i divides $u^{(2)}_{i,i} \cdot y_i$, and is primary with $u^{(2)}_{i,i}$ by definition of $U^{(2)}$, then, according to Gauss theorem, it also divides y_i. In other word we can find some $q'_1, \ldots, q'_p$ integers such that:

09/10/2004, EAST Version: 1.4.1

$$\begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ y_p \end{bmatrix} = \begin{bmatrix} q'_1 \cdot N_1 \\ \cdot \\ \cdot \\ q'_p \cdot N_p \end{bmatrix}$$

that is to say:

$$U^{(1)} \cdot \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_p \end{bmatrix} = \begin{bmatrix} q'_1 \cdot N_1 \\ \cdot \\ \cdot \\ q'_p \cdot N_p \end{bmatrix}$$

[0140] Now $U^{(1)}$ is a sub-diagonal matrix of integers with diagonal elements all equal to 1 or to -1. Then, as we have already shown it above, this involves that

$$\begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_p \end{bmatrix} = 0,$$

and since $(q_1, \ldots, q_p)$ were any integers, and $(x_1, \ldots x_p)$ any elements of $[-N_1+1, N_1-1] \times \ldots \times [-N_p+1, N_p-1]$, this involves that φ_U is injective, and therefore bijective, thus concluding the proof.

<u>ELEMENTARY FACTOR MERGE φ_n</u>

[0141] In order to get back to

$$S^{(k)} = \boxed{N}$$

in the end, some × product of sets must also sometimes be merged. The elementary factor merge φ_n function merges the two first factors of the first term of $S^{(n-1)}$. In fact the same principle could be used on any pair of factors of any term that have at least two factors, but this is not needed for the definition thanks to the term and factor permutation functions defined above.
In that case:

- $q_n = q_{n-1}$, i.e. same number of terms
- $p_{1,n-1} \geq 2$ and $p_{1,n} = p_{1,n-1} - 1$, i.e. one factor fewer in first term
- $\forall i \in \{2, \ldots, q_n\}\ T_i^{(n)} = T_i^{(n-1)}$, i.e. terms other than the first one are unchanged
- $\forall j \in \{3, \ldots, p_{1,n-1}\}\ N_{1,j-1}^{(n)} = N_{1,j}^{(n-1)}$, i.e. the $p_{1,n-1}-2$ last factors of $T_1^{(n-1)}$ are unaffected
- A and B defined by $A \triangleq N_{1,1}^{(n-1)}$, $B \triangleq N_{1,2}^{(n-1)}$, and thus $A \cdot B = N_{1,1}^{(n)}$
- a bijection

$$h: \quad \boxed{A} \times \boxed{B} \rightarrow \boxed{A \cdot B}$$
$$(x_1, x_2) \mapsto h((x_1, x_2))$$

φ_n is built as:

$$\varphi_n:\quad S^{(n-1)} \rightarrow S^{(n)}$$

$$x \mapsto \begin{cases} (1,(h(x_1, x_2), x_3\dots, x_{p_{1,n-1}})) & \text{if } t=1 \\ x & \text{otherwise} \end{cases}$$

with

$x = (t,v)$

and

$v = (x_1, x_2 \dots x_{p_{1,n-1}})$

[0142] The h bijections used by the elementary merge function can be defined as follows:

<u>$h: \boxed{A}\times\boxed{B}\rightarrow\boxed{A\cdot B}$ MAIN FACTOR MERGE FUNCTIONS</u>

[0143] There are many possible functions that all are computable on any current processor.

[0144] Here is one example:

In the following for all x and y integer such that y>0:

- x div y stands for the quotient of Euclidian division of x by y,
- x mod y stands for the remainder of Euclidian division of x by y.

Note that $\forall x\ \forall y > 0$, $(x \bmod y) \in \boxed{y}$, even for $x \leq 0$

The DIV-MOD factor merge function:

$$h:\quad \boxed{A}\times\boxed{B} \rightarrow \boxed{A\cdot B}$$

$$(x_1,x_2) \longmapsto B\cdot x_1+x_2$$

[0145] This function is clearly bijective and its inverse function is the div-mod factor split g bijection exemplified in the definition of the elementary factor split φ_n function:

[0146] Note: it is not needed to define a mod-mod factor merge h function that would be the inverse of the mod-mod factor split g function defined for the elementary factor split φ_n, that is to say:

$$g(x) = (x \bmod A,\ x \bmod B)\ .$$

$$g^{-1}((a,b)) = b + (((a-b)\cdot B') \bmod A)\cdot B$$

[0147] As a matter of fact the φ_n that would result from this mod-mod factor merge h would be the composition of an affine-mod φ_n of matrix defined by block :

09/10/2004, EAST Version: 1.4.1

$$U=\overbrace{\left[\begin{array}{c:c} \mathrm{Id} & O_V \\ \hdashline O_H & \begin{array}{cc} B' & 0 \\ 0 & 1 \end{array} \end{array}\right]}^{U^{(2)}} \bullet \overbrace{\left[\begin{array}{c:c} \mathrm{Id} & O_V \\ \hdashline O_H & \begin{array}{cc} 1 & -1 \\ 0 & 1 \end{array} \end{array}\right]}^{U^{(1)}}$$

and of null vector V, and of the elementary factor merge function using the div-mod factor merge h function.

[0148] Above Id, O_V, and O_H are respectively the identity (m-2)×(m-2) matrix (diagonal of 1), the null (m-2)×2 matrix and the null 2×(m-2) matrix.

ELEMENTARY TERM MERGE φ_n

[0149] In order to get back to $\boxed{N}$ in the end, some $\uplus$ sum of sets must also sometimes be merged. The elementary term merge φ_n function merges the two first factors respectively of the two first terms of $S^{(n-1)}$, when the two first terms have the same subsequent factors, that can therefore be factorised. In fact the same principle could be used on any pair of factor of any pair of terms in $S^{(n-1)}$, provided that they can be factorised in the form

$$(\boxed{A}\uplus\boxed{B})\times\boxed{A_1}\times\ldots\times\boxed{A_p}.$$

[0150] In that case:

- $q_{n-1}\geq 2$ and $q_n = q_{n-1}-1$, i.e. one fewer term
- $\forall j \in \{3,\ldots,p_{n-1}\}\ T^{(n)}_{j-1} = T^{(n-1)}_j$, i.e. remaining terms unchanged
- $p_{1,n-1} = p_{2,n-1} = p_{1,n}$, i.e. the two first terms of $S^{(n-1)}$ have the same number of factors, that is also that of the first term of $S^{(n)}$
- $\forall j \in \{2,\ldots,p_{1,n-1}\}\ N^{(n-1)}_{1,j} = N^{(n-1)}_{2,j}$, that is to say the two first term can be factorised
- A and B defined by $A \triangleq N^{(n-1)}_{1,1}$, $B \triangleq N^{(n-1)}_{2,1}$, and thus $A+B = N^{(n)}_{1,1}$
- a bijection κ: $\boxed{A}\uplus\boxed{B}\rightarrow\boxed{A+B}$

φ_n is built as:

09/10/2004, EAST Version: 1.4.1

$$\varphi_n: \quad S^{(n-1)} \rightarrow S^{(n)}$$

$$x \mapsto \begin{cases} (1, \kappa((t,x_1)), x_2, \ldots, x_{p_{1,n-1}}) & \text{if } t=1 \text{ or } t=2 \\ (t-1, v) & \text{otherwise} \end{cases}$$

With

$x = (t,v)$

and

$v = (x_1, x_2 \ldots x_{p_{1,n-1}})$

[0151] The term merge κ bijections used by the elementary term merge function can be defined as follows:

$$\underline{\kappa: \boxed{A} \uplus \boxed{B} \rightarrow \boxed{A+B}}\ \underline{\text{MAIN TERM MERGE FUNCTIONS}}$$

[0152] There are many possible functions that all are computable on any current processor.

[0153] As an example of κ, the mod-threshold term merge function will be described hereafter:

[0154] In this function, a merge function is made that is the inverse of the mod-threshod term split function that was already explained.

[0155] Similarly we have two constants C and T such that:

$$0 < T < C$$

$$\kappa: \quad \boxed{A} \uplus \boxed{B} \rightarrow \boxed{A+B}$$

$$(t,v) \mapsto x \text{ defined by the algorithm below}$$

[0156] Three constants x_M, v_M and t_M are derived from A, B and C according to the following definitions and formulas (the formulas are the same as for the mod-threshold term splitting function):

x_M is the greatest x while neither $\boxed{A}$ nor $\boxed{B}$ are empty.
t_M is the tag of the set ($\boxed{A}$ or $\boxed{B}$) not yet empty.
v_M is the greatest value of the set not yet empty while both sets are not yet empty.

09/10/2004, EAST Version: 1.4.1

$$x_M \triangleq \max\left\{ x \in \boxed{A+B} \;\Big/ \begin{cases} l\cdot T + c < A & \text{if } c < T \\ \text{and} & \\ l\cdot(C-T) + c - T < B & \text{otherwise} \end{cases}, \text{ where } l = x \text{ div } C, \text{ and } c = x \bmod C \right\}$$

$$t_M \triangleq \begin{cases} 1 & \text{if } ((x_M+1) \bmod C) \geq T \\ 2 & \text{otherwise} \end{cases}$$

$$v_M \triangleq \begin{cases} \max\left\{ v \;\Big/ \exists x \leq x_M, \begin{pmatrix} l \triangleq x \text{ div } C \\ \text{and} \\ c \triangleq x \bmod C \end{pmatrix} \Rightarrow \begin{pmatrix} c < T \\ \text{and} \\ v = l\cdot T + c \end{pmatrix} \right\} & \text{if } t_M = 1 \\ \max\left\{ v \;\Big/ \exists x \leq x_M, \begin{pmatrix} l \triangleq x \text{ div } C \\ \text{and} \\ c \triangleq x \bmod C \end{pmatrix} \Rightarrow \begin{pmatrix} c \geq T \\ \text{and} \\ v = l\cdot(C-T) + c - T \end{pmatrix} \right\} & \text{if } t_M = 2 \end{cases}$$

[0157] Then the algorithm that maps (t,v) onto x is the following:

```
if v ≤ v_M or t≠t_M then
   begin
      remark Neither [A] nor [B] are yet empty, then discrimination is
            based on tag
      if t = 1 then
         x := (v div T)·C + (v mod T);
      else
         x := T + (v div (C-T))·C + (v mod (C-T));
   end
else
   begin
      remark [A] or [B] is already empty, then no more discrimination, the
            remainder of the set not yet empty is mapped to [A+B]
      x := v - v_M + x_M;
   end
```

[0158] IMPLEMENTATION MATTERS

[0159] In fact the method of the invention is not tied to any particular implementation scheme. The method only defines mathematically a permutation, with a definition allowing less tabulation at the expense of more processing.

09/10/2004, EAST Version: 1.4.1

[0160] As was already suggested above, the computation of I^{-1} from its $\varphi_K \circ \varphi_{K-1} \ldots \circ \varphi_1$ definition might be done by a general purpose processor, typically during connection time, in order to prepare a look-up table to be used during interleaving. But as well, this computation might also be done by dedicated hardware circuitry, computing on the fly $I^{-1}(i)$ from its $\varphi_K \circ \varphi_{K-1} \ldots \circ \varphi_1$ definition in order to interleave the $(i+1)^{th}$ symbol of data when this symbol is to be interleaved. Typically a hardware circuitry could use a pipeline architecture where each step in the pipeline more or less correspond to one φ_n, or one compound $\varphi_{n+p} \circ \ldots \circ \varphi_n$. This kind of architecture is applicable because in order to compute on the fly $I^{-1}(i)$, what matters is not the total computation time of I^{-1} (i), but the rate at which they can be computed. This is because the "i" arguments of $I^{-1}(i)$ come in a predefined order.

[0161] When a hardware architecture is used, the operations implemented are not necessarily using the mathematical definition of φ_n. In order to reduce the circuit size, there can be a mixture of arithmetic units, and of ROM tables. For instance a first $\times$ splitting step can be done arithmetically, then the intermediate steps can be done by look up table, and a final $\times$ merging step can be done arithmetically again.

[0162] When a general purpose processor is used, neither are the operations implemented necessarily directly using the mathematical definition of φ_n. There might be some optimisations :

[0163] For instance when N is a power of 2, then splitting $\boxed{N}$ into a product of sets whose respective element numbers are also power of two does not need any division or modulo computation since it can be done by simple bit manipulation such as ORs, ANDs and bit shifts. For instance when mapping $\boxed{256}$ to $\boxed{8} \times \boxed{32}$ by the div-mod factor split function, then the image of x is (y,z), where if x=b7b6b5b4b3b2b1b0 in binary notation, then y=b7b6b5 and z=b4b3b2b1b0. To take it in more practical terms, if x=10110001, then y=101 and z=10001.

[0164] Also still in the case of a number N of elements that is a power of two, and when there is only one term (or when only one term of size N in the $\uplus$ sum is considered), some factor permutations are implementable by classical processing units instructions such as bit rotation or quartet swap. Let us for instance consider for N=256 the 16×16 rectangular interleaver. The 16×16 rectangular interleaver comprises in its φ_n compound definition a factor permutation that is swapping the two $\boxed{16}$ factors. In fact, the interleaver permutation can be shown to be not much more than a quartet swap which can be done in one instruction on some processing machines.

NON-UNIQUENESS OF DEFINITION

[0165] The inventive method provides means to define an interleaver using less tabulation at the expense of more processing, so that to reduce ROM size requirement.

[0166] The definition of the interleaver provided by the method might be not unique, since for instance:

- consecutive splitting steps can sometimes be done in several different ways; for instance a first way to map $\boxed{140}$ onto $\boxed{7} \times \boxed{4} \times \boxed{5}$ is a first div-mod splitting function φ_1 mapping $\boxed{140}$ onto $\boxed{7} \times \boxed{20}$, followed by a second div-mod split $\varphi_4 \circ \varphi_3 \circ \varphi_2$ mapping $\boxed{20}$ onto $\boxed{4} \times \boxed{5}$ (φ_2 and φ_4 are the factor permutation function respectively bringing $\boxed{20}$ in first position, and then $\boxed{7}$ in first position again); a second way is a first div-mod splitting function φ'_1 mapping $\boxed{140}$ onto $\boxed{28} \times \boxed{5}$, followed by a second div-mod split φ'_2 mapping $\boxed{28}$ onto $\boxed{7} \times \boxed{4}$; *both ways yield the same mapping* ($\varphi_4 \circ \varphi_3 \circ \varphi_2 \circ \varphi_1 = \varphi'_2 \circ \varphi'_1$),
- the order of compound of two functions whose action is limited to different factor or terms can be inverted,
- it is possible to build a function φ_n and its inverse by the inventive method, then you can always insert a compound of several φ_n into the compound defining I^{-1}, provided that the compound of these several φ_n sums up to the identical function, and has no overall effect, for instance when $I^{-1} = \varphi_K \circ \ldots \circ \varphi_2 \circ \varphi_1$ is an interleaver over $\boxed{N}$, and $N = L \cdot C$, then if φ_{K+1} is defined as the div-mod factor split function that maps $\boxed{N}$ to $\boxed{L} \times \boxed{C}$ and φ_{K+2} as the div-mod factor merge function that maps $\boxed{L} \times \boxed{C}$ to $\boxed{N}$, then I^{-1} can also be defined as $I^{-1} = \varphi_{K+2} \circ \varphi_{K+1} \circ \varphi_K \circ \cdots \circ \varphi_2 \circ \varphi_1$ because the compound $\varphi_{K+2} \circ \varphi_{K+1}$ has no overall effect.
- the classification of the φ_n might not be unique, some φ_n can be found under two categories. For instance when all the $N_{1j}^{(n)}$ are equal then a "factor permutation" φ_n can also be defined as an "affine-mod" φ_n with V=0 and U a permutation matrix.

EXAMPLE OF INVENTIVE METHOD APPLIED TO KNOWN INTERLEAVERS

[0167] The inventive method allows to build interleavers by mixing tabulation (or processing not defined in the invention) and processing (defined in the invention). Here is shown that some well known interleavers can be defined with the inventive method to be implemented in an interleaver device according to the invention.

1 - THE RECTANGULAR INTERLEAVER

[0168] According to the invention, the classical rectangular interleaver I of depth $N = L \cdot C$ with L lines and C column

block is written as:

$$I^{-1}=\varphi_3{}^\circ\varphi_2{}^\circ\varphi_1$$

Where:

- φ_1 is the div-mod elementary factor split function splitting $\boxed{N}$ to $\boxed{L}\times\boxed{C}$
- φ_2 is the factor permutation function for $\sigma=(2,1)$, that is to say φ_2 is mapping $\boxed{L}\times\boxed{C}$ on $\boxed{C}\times\boxed{L}$
- φ_3 is the div-mod elementary factor merge function merging $\boxed{C}\times\boxed{L}$ to $\boxed{N}$ The rectangular interleaver is illustrated on figure 6.

2 -THE DIAGONAL INTERLEAVER

[0169] The usual diagonal interleaver I depth $N=L\cdot C$, with L lines and C column block, where lines are written first and then diagonal are read, when j^{th} diagonal begins at the first element of j^{th} line is written as:

$$I^{-1}=\varphi_3{}^\circ\varphi_2{}^\circ\varphi_1$$

Where:

- φ_1 is the div-mod elementary factor split function splitting $\boxed{N}$ to $\boxed{L}\times\boxed{C}$
- φ_2 is the affine-mod function with V=0 and

$$U=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}$$

- φ_3 is the div-mod elementary factor merge function merging $\boxed{L}\times\boxed{C}$ to $\boxed{N}$

3 - THE TRUNCATED RECTANGULAR INTERLEAVER

[0170] Truncated rectangular interleavers are used in MIL interleavers which are defined in ETSI SMG2 UMTS-L1 EG Tdoc 98/273 "Multi-stage interleaving (MIL) method for Turbo codes" by "NTT DoCoMo". In a truncated rectangular interleaver the N input data is written along lines in a block of L' lines and C' columns, with C' which is not a divider of N (and L' great enough for $L'\cdot C'\geq N$). Then the data is read out along columns.

[0171] The truncated rectangular interleaver is illustrated on figure 7 with N = 20 and C' = 6.

[0172] The truncated rectangular interleaver is defined by the method of the invention as the $\uplus$ sum of two rectangular interleavers.

$$I^{-1}=\varphi_8{}^\circ(\varphi_7{}^\circ\varphi_6{}^\circ\varphi_5)^\circ(\varphi_4{}^\circ\varphi_3{}^\circ\varphi_2)^\circ\varphi_1$$

Where :

- φ_1 is the mod-threshold elementary term split function, where C = C' and T = N mod C'.
  φ_1 $\boxed{N}$ to $\boxed{N_1}\uplus\boxed{N_2}$ with

$$N_1 = (N \text{ div } C'+1)\cdot(N \text{ mod } C'), \text{ and}$$

$$N_2 = (N \text{ div } C')\cdot(C'-(N \text{ mod } C'))$$

- $(\varphi_4{}^\circ\varphi_3{}^\circ\varphi_2)$ is the first rectangular interleaver that is applied to $\boxed{N_1}$ with parameters:

$$L = (N \text{ div } C'+1), \text{ and}$$

$$C = (N \text{ mod } C')$$

09/10/2004, EAST Version: 1.4.1

- $(\varphi_7 \circ \varphi_6 \circ \varphi_5)$ is the second rectangular interleaver that is applied to $\boxed{N_2}$ with parameters:

$$L = (N \text{ div } C'), \text{ and}$$

$$C = C' - (N \text{ mod } C')$$

- φ_8 is the term merge function with $C = N_1+N_2$ and $T = N_1$.

4 -THE MIL INTERLEAVER

[0173] The MIL interleavers defined in ETSI SMG2 UMTS-L1 EG T doc 98/273 "Multi-stage interleaving (MIL) method for Turbo codes" by "NTT DoCoMo" can also be constructed by the method of the invention. The MIL interleaver is defined recursively, and the method of the invention is by essence recursive.

[0174] A MIL interleaver of depth $N = L \cdot C$ is defined as the succession of the following steps:

1) First write the input data along the lines of the rectangular block of L lines and C columns;
2) Then interleave each line by either a rectangular interleaver, or a trunctated rectangular interlaver, or a MIL interleaver; and
3) Finally read the block by columns.

[0175] The first step consists of $(\varphi_2 \circ \varphi_1)$, where φ_1 is the div-mod factor split function applied to $\boxed{N}$ and mapping it to $\boxed{L} \times \boxed{C}$, and φ_2 is the distribution function that maps $\boxed{L} \times \boxed{C}$ to

$$\underbrace{\boxed{C} \uplus \ldots \uplus \boxed{C}}_{\text{L times}},$$

where each term consists of a line of the block.

[0176] The second steps consists in applying a rectangular, a truncated rectangular or a MIL interleaver to each term. As already seen, rectangular and truncated rectangular interleaver can be generated by the method of the invention, then this second step is also within the invention.

[0177] The third step consisting of reading the block by column can be simply written as $(\varphi_k \circ \varphi_{k-1} \circ \varphi_{k-2})$, where φ_{k-2} is the factorisation function mapping

$$\underbrace{\boxed{C} \uplus \ldots \uplus \boxed{C}}_{\text{L times}}$$

to $\boxed{L} \times \boxed{C}$, φ_{k-1} a factor permutation mapping $\boxed{L} \times \boxed{C}$ to $\boxed{C} \times \boxed{L}$, and φ_k the div-mod factor merge.

[0178] In conclusion MIL interleavers are a subset of the interleavers that can be generated by the inventive method.

5 - OTHER INTERLEAVERS

[0179] The inventive method is open to embed other interleavers into the inventive interleavers by means of the elementary interleaving function. After decomposing

$$S^{(0)} = \boxed{N}$$

into a sum of products of sets of the $\boxed{X}$ form, an interleaver not defined by this document can be applied to only one of this factor set : this is what is called elementary interleaving in this document.

[0180] In the definition of many existing interleaver a similar method is also used, for instance in ETSI SMG2 UMTS-L1 EG Tdoc 98/337 "A General Turbo Interleaver Design Technique with Near Optimal Performance " by " Hughes Network Systems", we can read that such a process is used :

09/10/2004, EAST Version: 1.4.1

- writing the symbol in a rectangular block along the rows: this corresponds to a div-mod factor split of $\boxed{N}$ to $\boxed{L}\times\boxed{C}$
- then operating on each row separately: this corresponds to a distribution of $\boxed{L}\times\boxed{C}$ product in a sum of L terms $\oplus \ldots \oplus \boxed{C}$, each term being a row,
- shuffling each row by some rule derived from Galois field arithmetic: this corresponds to an elementary interleaving applied to the term of which the row is consisting.
- permutating the rows : this can be expressed either as a term permutation, or as factorising to $\boxed{L}\times\boxed{C}$ again, and making an elementary interleaving bearing on the $\boxed{L}$ dimension.
- reading by column: this is also interpreted in the inventive method, we first swap the two factors in $\boxed{L}\times\boxed{C}$ by a factor permutation, then make a div-mod factor merge.

[0181] The invention brings some methodology allowing to define an interleaver in a unified formal language. This would ease specification of interleavers by defining the compound or sequence of the φ_n. Such a specification would be without ambiguity, and even possibly in a language that is directly machine processable. Such a sequence can be coded in a dedicated binary format, for instance a TLV format, where T (tag) gives the type of φ_n (factor split, term split, affine-mod, etc...) and the LV (length+value) gives the characteristics of φ_n, for instance if φ_n is affine-mod, the matrix U and the vector V. Such a coded format can be used to defined a great variety of interleavers in a device at a very low ROM memory cost. Also the same format can be used on an interface to negotiate which interleaver is to be used.

[0182] The term split is also of a great interest. Many arithmetic interleavers (e.g. GF interleaver as defined in ETSI SMG2 UMTS-L1 EG Tdoc 98/337 "A General Turbo Interleaver Design Technique with Near Optimal Performance " by" Hughes Network Systems", for $N=2^m$) are primarily defined for some N that has good properties. When an interleaver with an N that has not such good properties needs to be built, then an arithmetic interleaver with N'>N and N' having the good properties is built, and then punctured down to N. Puncturing an interleaver is not done at a very high processing power when it consists in modifying a RAM table. However these forces an implementation where first the interleaver is prepared in a RAM table, and then this RAM table is used. Implementation where the interleaver function is computed real time by an ASICs cannot be realistically considered with puncturing. Thanks to the term splitting function instead of searching for N' with good arithmetic properties and N'>N, one can find N' with good arithmetic properties such that N'<N, and then apply the arithmetic interleaver to a term of size N', thus avoiding any puncturing.

[0183] Puncturing of an interleaver is shown on figure 8. An interleaver I_8 of depth 8 is punctured down to a depth of 5 to form an interleaver I_5. In the upper part of the figure a table where the values of I_8^{-1} are shown. The table entry number (above the table) is the value of i, and the entry value (inside the table) is the value of $I_8^{-1}(i)$. In other words we have $I_8^{-1}(0)=3$, $I_8^{-1}(1)=0$, etc.... Then all the entries such that $I_8^{-1}(i)\geq 5$ are to be punctured. These entries are marked on the figure by a triangle. Puncturing consists in moving some part of the table, for instance the entry in position 4 is moved to position 2 so that $I_5(2) = I_8(4) = 1$.

[0184] After puncturing, the 5 first entries of the table contain only numbers ranging from 0 to 4.

Claims

1. Interleaver device for interleaving a data set comprising:

   - processing unit (10) including a data processor (16) for running an interleaver (I^{-1}) under the control of driving means (18);
   - input means (12) for inputting the data set to be interleaved and output means (14) for outputting the interleaved data set;

   characterised in that:
   said driving means (18) include:

   mapping processing means (20) for performing a set of bijective elementary functions (φ_n) and supplying a mapping of the interleaver to the data processor (16) for interleaving the data set according to this mapping; and
   interleaver definition means (24) for supplying said mapping processing means (20), with a definition of said interleaver (I^{-1}), expressed as a compound function ($\varphi_K o \ldots o \varphi_1$) of elementary functions (φ_n), each elementary functions coming from said set of bijective elementary functions (φ_n), for said mapping processing means (20) to perform each of said functions, compounded according to the interleaver definition (I^{-1}), and so providing to the data processor (16) said mapping according to which the data processor (16) interleaves the data set and supply the interleaved data set to the output means (14).

2. Interleaver device according to claim 1, characterised in that the driving means (18) includes a read only memory (22) from which the set of the bijective elementary functions (φ_n) can be derived.

09/10/2004, EAST Version: 1.4.1

3. Interleaver device according to claim 1 or 2, characterised in that the interleaver definition means (24) comprise means for receiving the interleaver definition (I^{-1}).

4. Interleaver device according to any one of the preceding claims, characterised in that said set of elementary functions (φ_n) includes at least the following functions whose origin and destination sets can both be expressed as a $\uplus$ sum of $\times$ products of factor sets that all are sets of successive integers:

- an elementary term split function, which is splitting, into a $\uplus$ sum of two sets, a factor of a term of a set;
- an elementary factor split function, which is splitting, into a x product of two sets, a factor of a term of a set;
- an elementary permuting function, which is permuting the elements of a factor of a term of a set;
- an elementary affine-mod function, which is acting on a term of a set as if it was a subset of a vector space whose dimension is the number of factors, by an affine mapping, followed by taking the modulus of each coordinate by the size of the respective factor set;
- an elementary factor merge function, which is merging a x product of sets, into a set by merging two factors of a term;
- an elementary term merge function, which is merging a $\uplus$ sum of sets into a set by merging two factors respectively of two terms of a set, when the two terms have the same other factors.

5. Interleaver device according to claim 4, characterised in that said set of elementary functions (φ_n) further includes the following functions whose origin and destination sets can both be expressed as a $\uplus$ sum of $\times$ product of factor sets that all are sets of successive integers:

a elementary term permutation function ; and
a elementary factor permutation function;
and in that:

- the elementary term split function is splitting, into a $\uplus$ sum of two sets, the first factor of the first term of a set;
- the elementary factor split function is splitting, into a x product of two sets, the first factor of the first term of a set;
- the elementary permuting function is permuting the elements of the first factor of the first term of a set;
- the elementary factor merge function is merging a x product of sets into a set, by merging the two first factors of the first term of a set;
- the elementary term merge function, which is merging a $\uplus$ sum of sets a sets into a set, by merging the two first factors respectively of the two first terms of a set, when the two first terms have the same subsequent factors.

6. Interleaver device according to claims 4 or 5, characterised in that the elementary factor split function is splitting one factor $\boxed{A \cdot B}$ of a term in the origin set by the following mapping:

$$\begin{array}{ccc} \boxed{A \cdot B} & \rightarrow & \boxed{A} \times \boxed{B} \\ x & \mapsto & (x \text{ div } B,\ x \bmod B) \end{array}$$

7. Interleaver device according to claims 4 or 5, characterised in that the elementary factor split function is splitting one factor $\boxed{A \cdot B}$ of a term in the origin set by the following mapping:

$$\begin{array}{ccc} \boxed{A \cdot B} & \rightarrow & \boxed{A} \times \boxed{B} \\ x & \mapsto & (x \bmod A,\ x \bmod B) \end{array}$$

8. Interleaver device according to anyone of claims 4 to 7, characterised in that the elementary factor merge function

09/10/2004, EAST Version: 1.4.1

is merging two factors $\boxed{A}$ and $\boxed{B}$ of a term in the origin set by the following mapping:

$$\begin{array}{ccc} \boxed{A}\times\boxed{B} & \rightarrow & \boxed{A\cdot B} \\ (x_1,x_2) & \mapsto & B\cdot x_1+x_2 \end{array}$$

9. Interleaver device according to any one of claims 4 to 8, characterised in that the elementary affine-mod function is defined by the affine function of matrix U which is the product $U = U^{(2)} \cdot U^{(1)}$ of two matrices $U^{(1)}=[u^{(1)}_{i,j}]$, and $U^{(2)}=[u^{(2)}_{i,j}]$ such that

- $U^{(1)}$ is a sub-diagonal matrix with a diagonal of 1 or -1, that is to say:

$$\begin{cases} \forall\ (i,j)\in \{1,\ldots,m\}^2,\ j>i \Rightarrow u^{(1)}_{i,j}=0 \\ \text{and} \\ \forall\ i\in \{1,\ldots,m\},\quad u^{(1)}_{i,i}=1 \text{ or } u^{(1)}_{ij}=-1 \end{cases}$$

- $U^{(2)}$ is a diagonal matrix with for all j, j^{th} diagonal element primary with $N^{(n)}_{i,j}$ that is to say:

$$\begin{cases} \forall\ (r,c)\in \{1,\ldots,m\}^2,\ r\neq c \Rightarrow u^{(2)}_{r,c}=0 \\ \text{and} \\ \forall\ j\in \{1,\ldots,m\},\quad u^{(2)}_{j,j} \text{ and } N^{(n)}_{ij} \text{ greatest common divider is } 1 \end{cases}$$

10. Interleaver device according to any one of claims 4 to 9, characterised in that the elementary term split function is splitting one factor $\boxed{A+B}$ into $\boxed{A} \uplus \boxed{B}$ by selecting the tag of the image based on a comparison to a constant threshold of the modulus of its $\boxed{A+B}$ input by a some fixed constant, at least for some of its $\boxed{A+B}$ input values.

11. Interleaver device according to any one of claims 4 to 10, characterised in that the elementary term merge function is merging into $\boxed{A+B}$ two factors $\boxed{A}$ and $\boxed{B}$ taken in terms with same respective other factors, such that its inverse bijection selects the tag of the image in $\boxed{A} \uplus \boxed{B}$ based on a comparison to a constant threshold of the modulus of its $\boxed{A+B}$ input by a some fixed constant, at least for some of its $\boxed{A+B}$ input values.

12. Method for interleaving a data set according to an interleaver (I^{-1}), characterised in that the interleaver (I^{-1}) is defined as a compound function ($\varphi_k o \ldots o \varphi_1$) of bijective elementary functions (φ_n), each elementary function (φ_n) coming from a set of bijective elementary functions (φ_n), and in that it includes the steps in which the elementary functions, compounded according to the interleaver definition (I^{-1}), are subsequently applied to the data set for providing the interleaved set of data.

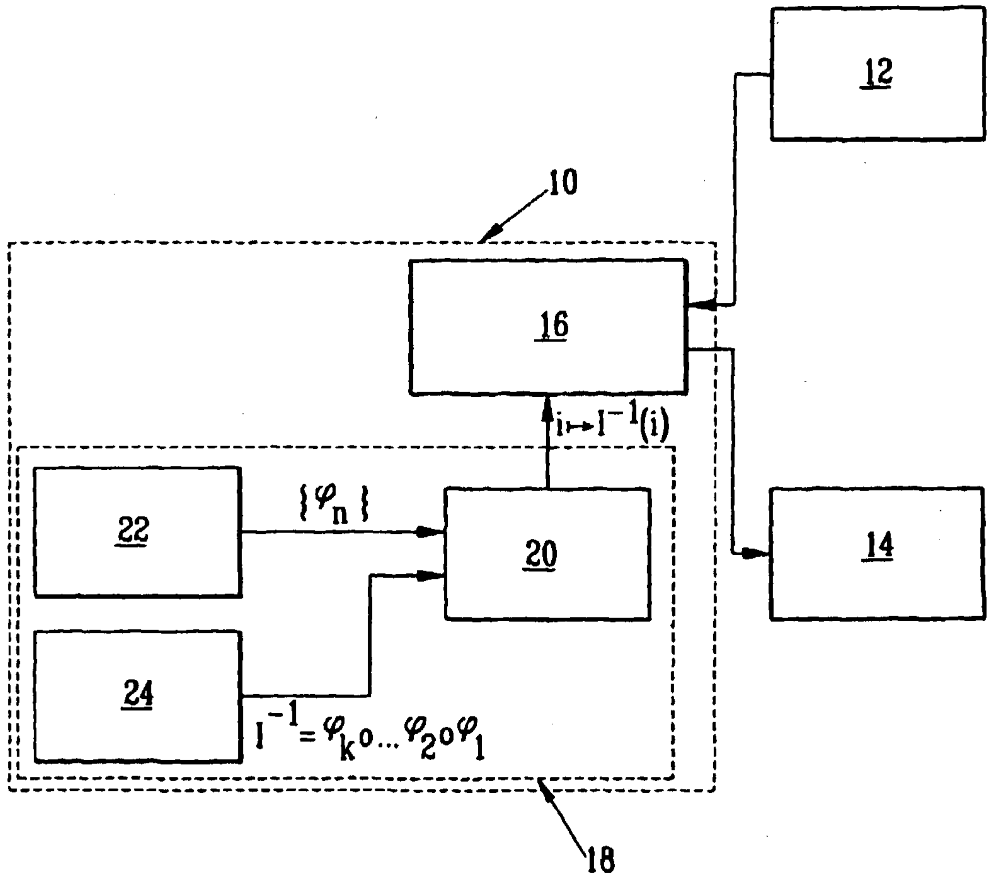


FIG.1

09/10/2004, EAST Version: 1.4.1

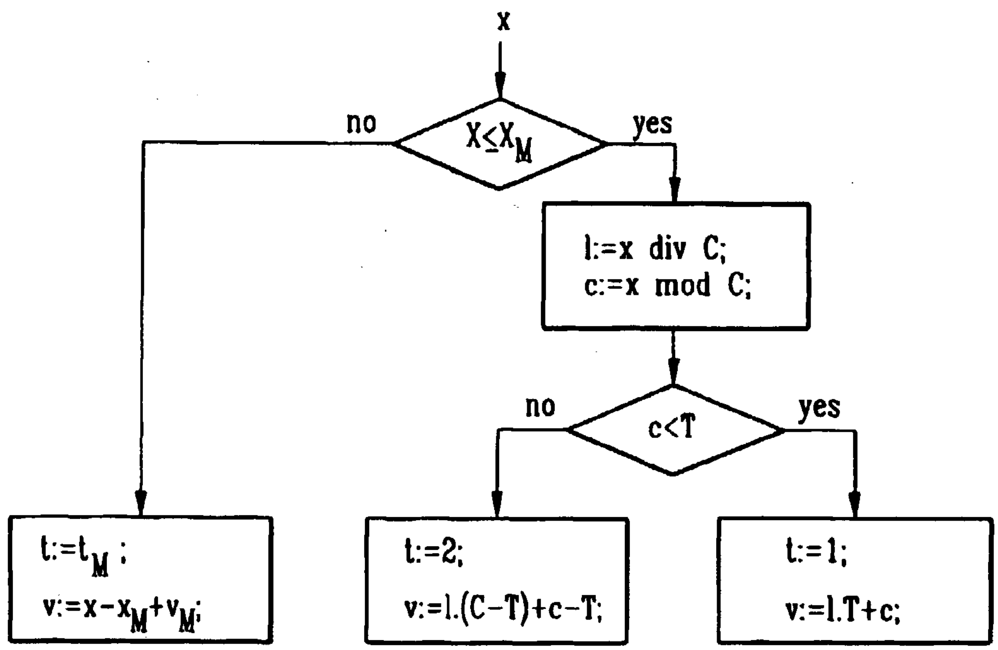


FIG.2

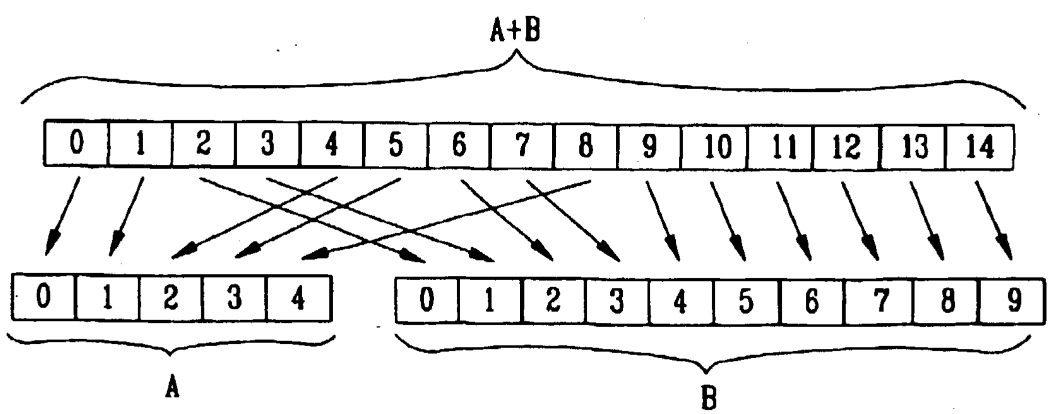


FIG.3

09/10/2004, EAST Version: 1.4.1

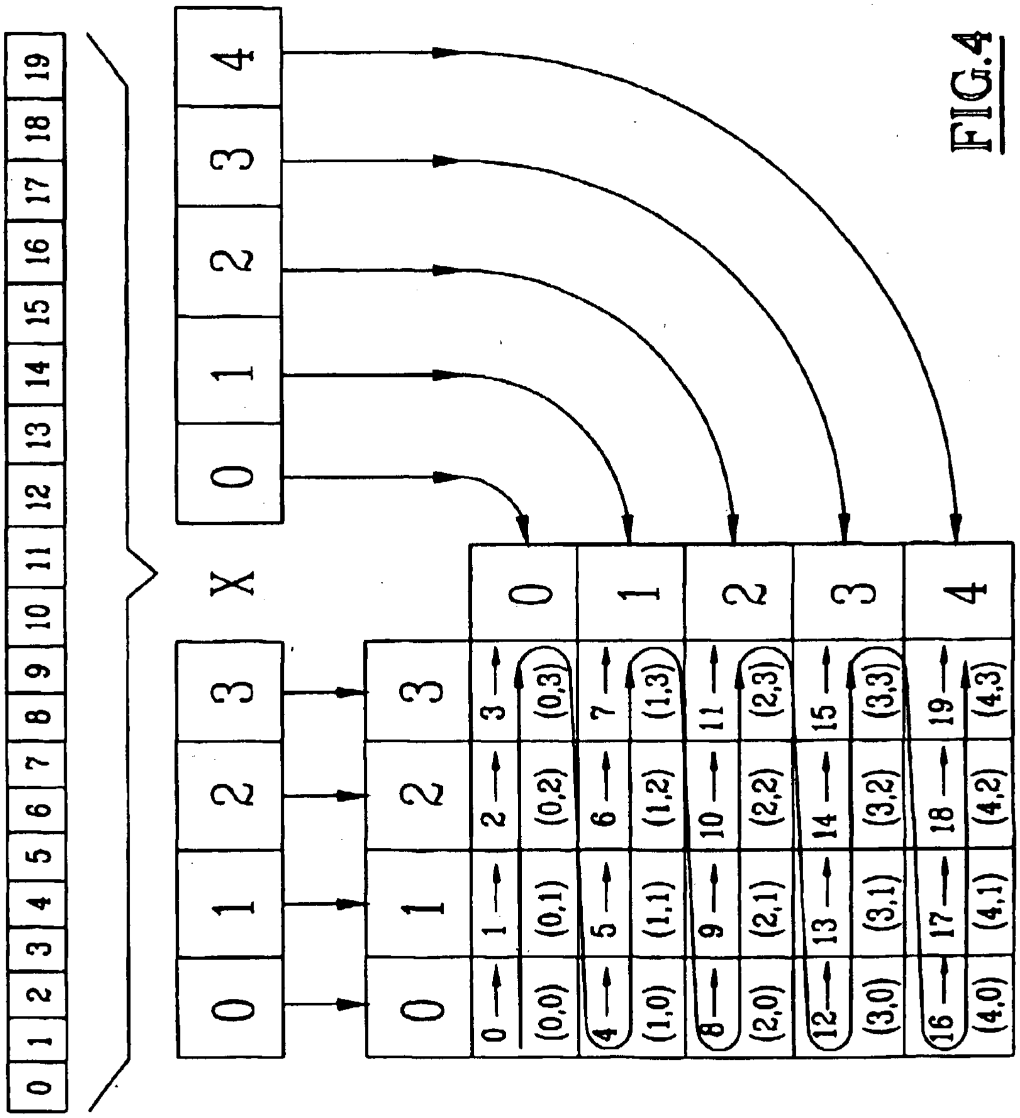
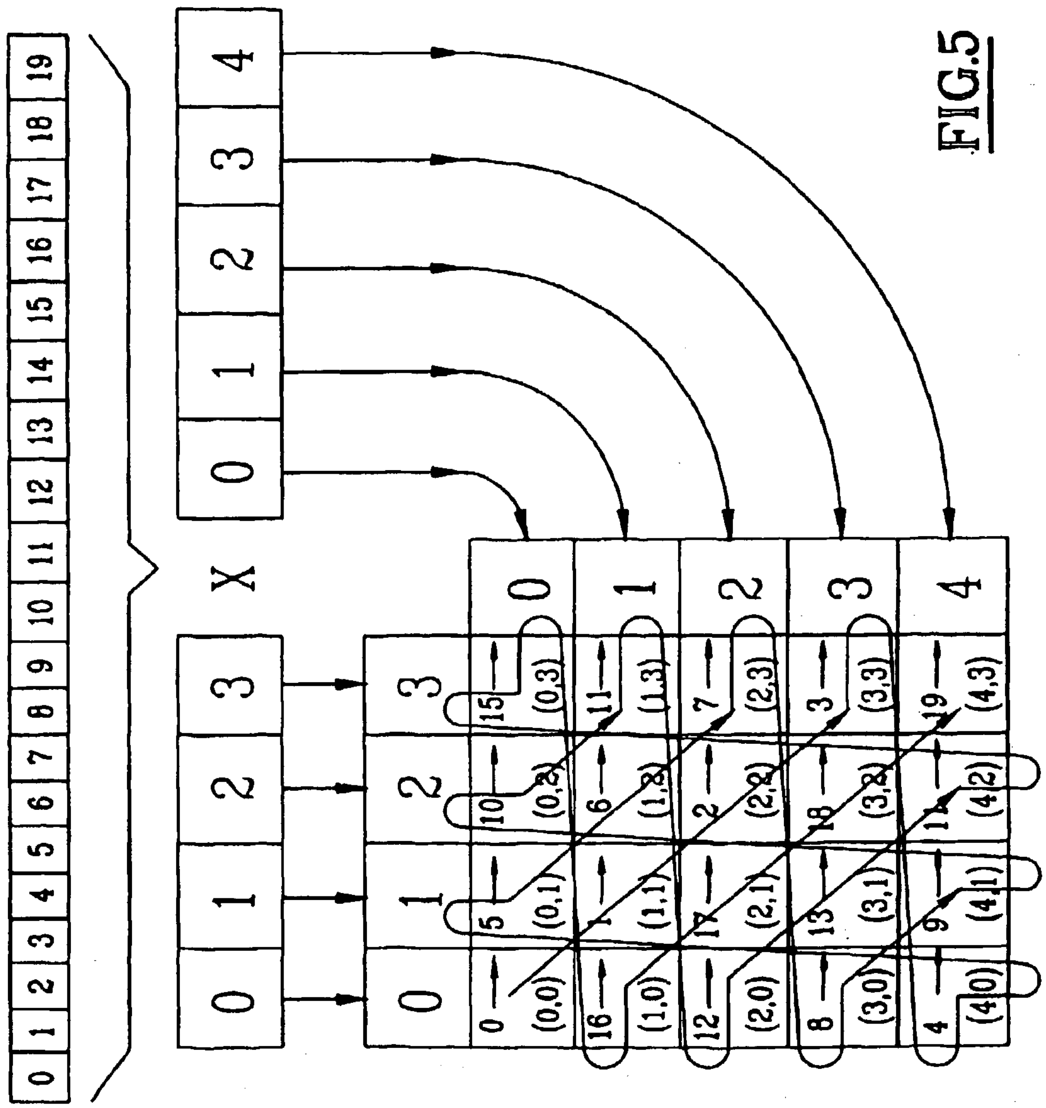


FIG.4

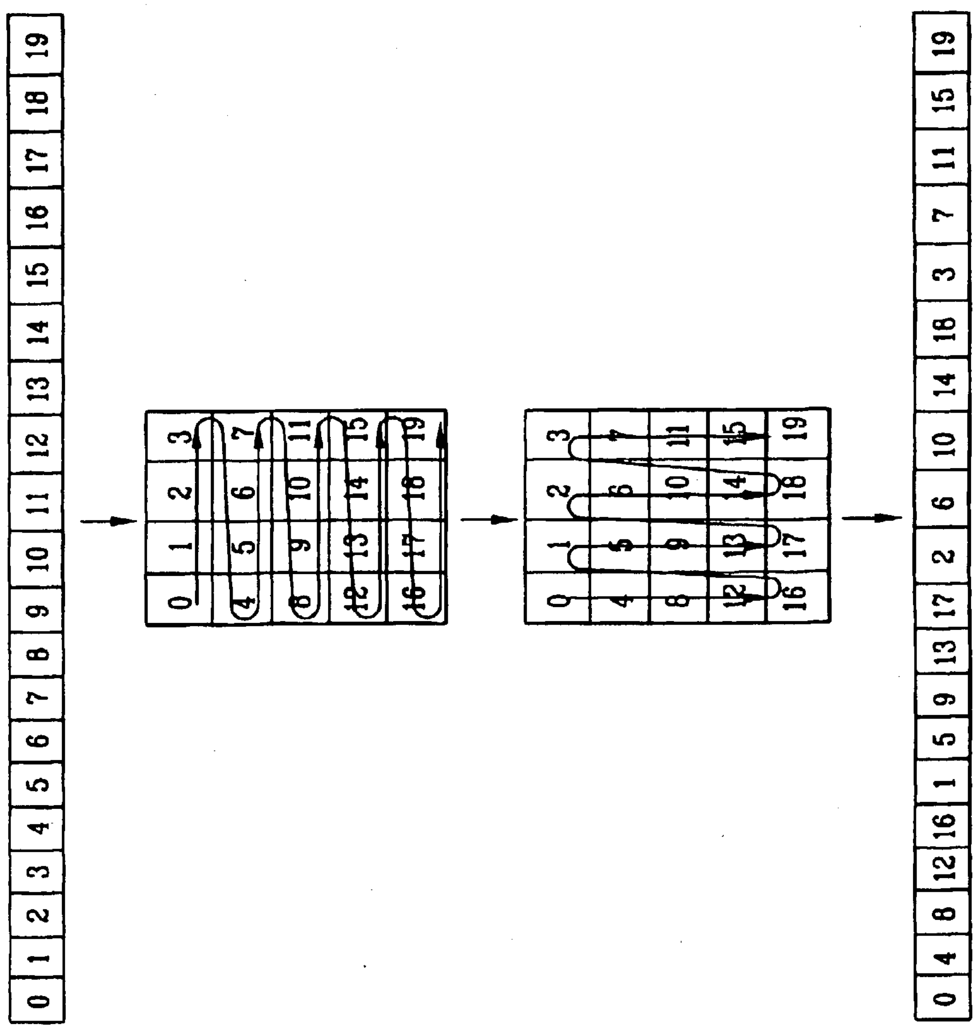
09/10/2004, EAST Version: 1.4.1



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
0 1 2 3
X
0 1 2 3 4
(0,0) (0,1) (0,2) (0,3)
(1,0) (1,1) (1,2) (1,3)
(2,0) (2,1) (2,2) (2,3)
(3,0) (3,1) (3,2) (3,3)
(4,0) (4,1) (4,2) (4,3)

FIG.5

FIG.6



09/10/2004, EAST Version: 1.4.1

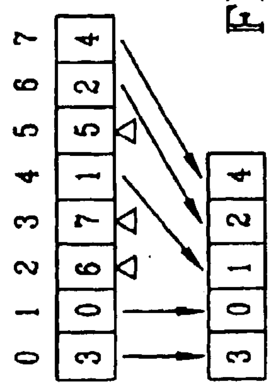


FIG.8

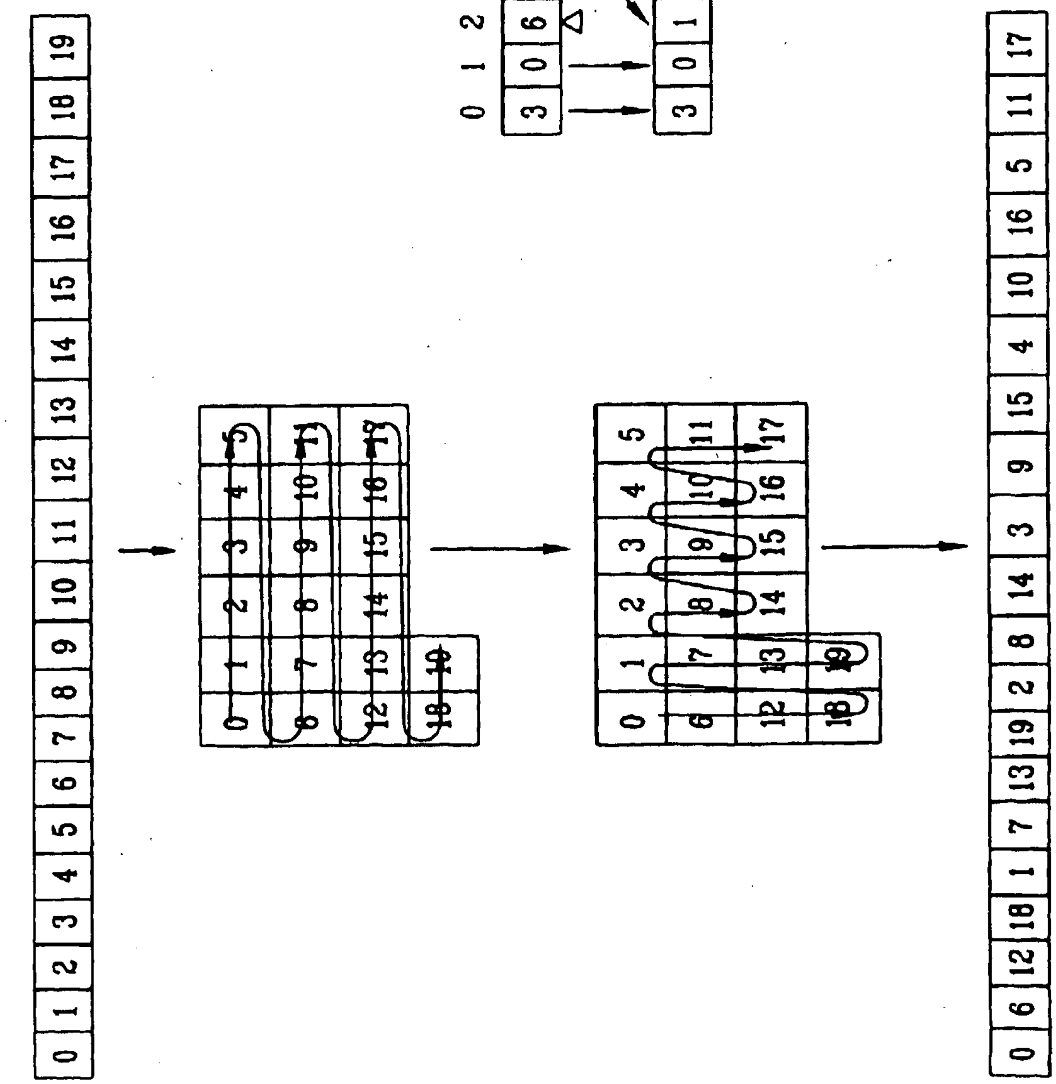


FIG.7

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 40 0537

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION
A	WO 95 18489 A (ZENITH ELECTRONICS CORP) 6 July 1995 (1995-07-06) * page 13, line 19 - page 19, line 19; figure 8 * ---	1-3	H03M13/27
A	US 5 136 588 A (ISHIJIMA TOMOHARU) 4 August 1992 (1992-08-04) * the whole document * ---	1,12	
A	WO 97 05702 A (FRANCE TELECOM ;TELEDIFFUSION FSE (FR); COMBELLES PIERRE (FR); CAS) 13 February 1997 (1997-02-13) * the whole document * ---	1,12	
A	FR 2 507 844 A (THOMSON CSF) 17 December 1982 (1982-12-17) ---		
A	TROUP T P ET AL: "ADAPTIVE DATA INTERLEAVING USING A MICROPROCESSOR CONTROLLED RECONFIGURABLE GATE ARRAY" MICROPROCESSING AND MICROPROGRAMMING, vol. 36, no. 1, 1 November 1992 (1992-11-01), pages 43-48, XP000330570 ISSN: 0165-6074 ---		TECHNICAL FIELDS SEARCHED
A	US 5 592 492 A (BEN-EFRAIM NADAV ET AL) 7 January 1997 (1997-01-07) ---		H03M
A	WO 96 37050 A (ADVANCED HARDWARE ARCHITECTURE) 21 November 1996 (1996-11-21) -----		

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	9 September 1999	Devergranne, C

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

09/10/2004, EAST Version: 1.4.1

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 40 0537

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

09-09-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9518489 A	06-07-1995	US 5572532 A	05-11-1996
		CN 1141101 A	22-01-1997
		EP 0737385 A	16-10-1996
		JP 9507352 T	22-07-1996
US 5136588 A	04-08-1992	JP 1037125 A	07-02-1989
		WO 8901265 A	09-02-1989
WO 9705702 A	13-02-1997	FR 2737363 A	31-01-1997
		DE 69601430 D	04-03-1999
		DE 69601430 T	15-07-1999
		EP 0840958 A	13-05-1998
		ES 2129273 T	01-06-1999
FR 2507844 A	17-12-1982	NONE	
US 5592492 A	07-01-1997	US 5745497 A	28-04-1998
WO 9637050 A	21-11-1996	NONE	

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

09/10/2004, EAST Version: 1.4.1